AD-A057 278

AFFDL-TR-78-18
Volume II

MONTE CARLO BAYESIAN SYSTEM RELIABILITY — AND MTBF-CONFIDENCE ASSESSMENT, II

Volume II: SPARCS-2 Users Manual

MITCHELL O. LOCKS
OKLAHOMA STATE UNIVERSITY
STILLWATER, OKLAHOMA 74074

MARCH 1978

TECHNICAL REPORT AFFDL-TR-78-18, Volume II
Final Report March 1976 – December 1977

Approved for public release; distribution unlimited.

AIR FORCE FLIGHT DYNAMICS LABORATORY
AIR FORCE WRIGHT AERONAUTICAL LABORATORIES
AIR FORCE SYSTEMS COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433

NOTICE

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

This report has been reviewed by the Information Office (OI) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.

This technical report has been reviewed and is approved for publication.

H. Leon Harter

H. LEON HARTER
Project Engineer

Richard D. Krobusek

RICHARD D. KROBUSEK, Major, USAF
Chief, Analysis & Optimization Branch

FOR THE COMMANDER

Holland B. Lowndes, Jr.

HOLLAND B. LOWNDES, JR.
Acting Chief, Structural Mechanics Division

"If your address has changed, if you wish to be removed from our mailing list, or if the addressee is no longer employed by your organization please notify AFFDL/FBRD, W-PAFB, OH 45433 to help us maintain a current mailing list."

Copies of this report should not be returned unless return is required by security considerations, contractual obligations, or notice on a specific document.

AIR FORCE/56780/6 June 1978 — 450

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER AFFDL-TR-78-18, Volume II	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) SPARCS-2 Users Manual		5. TYPE OF REPORT & PERIOD COVERED Final Report March 1976-December 1977
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Mitchell O. Locks and Ross Gregson		8. CONTRACT OR GRANT NUMBER(s) F33615-76-C-3094
9. PERFORMING ORGANIZATION NAME AND ADDRESS Oklahoma State University College of Business Administration Stillwater, OK 74074		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 61102F 2304N104
11. CONTROLLING OFFICE NAME AND ADDRESS Air Force Flight Dynamics Laboratory/FBRD Air Force Wright Aeronautical Laboratories Air Force Systems Command Wright-Patterson Air Force Base, Ohio 45433		12. REPORT DATE March 1978
		13. NUMBER OF PAGES 44
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

AFFDL-TR-78-18, Volume I, is titled "Monte Carlo Bayesian Reliability- and MTBF-Confidence Assessment, II

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Reliability	Poincaré's theorem	Conjugate prior distribution	Beta distribution
Monte Carlo	Complex systems		Negative-log gamma distribution
Assessment	Inclusion-exclusion	Modularization	
System assessment	Bernoulli process	Poisson process	Minimal paths

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This SPARCS-2 (Simulation Program for Assessing the Reliabilities of Complex Systems, Version 2) Users Manual provides the information and instructions for using the SPARCS-2 computer program. A glossary is included containing the general terminology and a description of the input terms. The manual also includes a general description of the program system capabilities and its operation as well as a listing of the specific functions performed by the program. In addition, sections containing usage instructions and control or operating

Block 19 (Continued)

Minimal cuts	Subsystems	Pass-fail components
Probability	Failure modes	Time-to-failure components
Simulation	Component assessment	
Logic	MTBF assessments	

Block 20 (Continued)

instructions are also included. These instructions cover areas such as: data input format, formalized deck structure, and the procedure for putting SPARCS on a load module and for executing the stored program. Two example cases are also included which demonstrate setting up a problem, the input data deck, and the resulting output of the analysis.

PREFACE

This users manual has been prepared as an aid to users of the SPARCS-2 computer program in assessing the reliability and/or the MTBF of complex systems, as discussed in Volume I of this report. Both volumes were prepared under Contract No. F33615-76-C-3094, which was technically monitored by Dr. H. Leon Harter. This work was performed under work unit 2304N104, System Reliability - Confidence Assessment.

TABLE OF CONTENTS

INTRODUCTION

This SPARCS-2 (Simulation Program for Assessing the Reliabilities of Complex Systems, Version 2) Users Manual provides the information and instructions for using the SPARCS-2 computer program. A glossary is included containing the general terminology and a description of the input terms. The manual also includes a general description of the program system capabilities and its operation as well as a listing of the specific functions performed by the program. In addition, sections containing usage instructions and control or operating instructions are also included. These instructions cover areas such as: data input format, formalized deck structure, and the procedure for putting SPARCS on a load module and for executing the stored program. Two example cases are also included which demonstrate setting up a problem, the input data deck, and the resulting output of the analysis.

SPARCS-2 is the result of a continuation of the research effort at Oklahoma State University for exact estimation and assessment of the reliabilities of complex systems, performed largely under the sponsorship of the Air Force Flight Dynamics Laboratory. The original version of SPARCS was programmed in 1975 by J. W. Cooley, and it is documented in both References [4] and [5]. Both SPARCS and SPARCS-2, being Monte Carlo programs, incorporate an estimation program MAPS (Method for the Analysis of the Probabilities of Systems) developed by J. L. Burris [1]. MAPS in turn is based on earlier FORTRAN programs developed at Rockwell International for NASA, documented in [2] and [3].

COMPUTER PROGRAM SYSTEM CAPABILITIES

Purpose

SPARCS-2 (Simulation Program for Assessing the Reliabilities of Complex Systems, Version 2) is a PL/1 computer program for assessing (establishing interval estimates for) the reliability and the MTBF of a large and complex system of any modular configuration. The system can consist of a complex logical assembly of independently failing attribute (binomial-Bernoulli) and time-to-failure (Poisson-exponential) components, without regard to their placement. Alternatively, it can be a configuration of independently failing modules, where each module has either or both attribute and time-to-failure components.

SPARCS-2 also has an improved "super modularity" feature. Modules with minimal-cut unreliability calculations can be mixed with those having minimal-path reliability calculations. All output has been standardized to system reliability or "probability of success", regardless of the form in which the input data is presented, and whatever the configuration of modules or elements within modules.

General Description

The raw data for the reliability assessments are the component failure history data and the system configuration. The historical data are "successes and failures" for binomial-Bernoulli components and "failures and testing time (normalized to 'mission equivalent units')" for time-to-failure components. The configuration data consist of a list or lists of minimal

paths ("minimal path sets" or "tie sets"), or else a list of minimal cuts ("minimal cut sets"), for the system as a list of modules, and for each module as a list of components. If the MTBF assessment option is selected, the system "mission time" is also needed.

The underlying mathematical model is an amalgamation of Boolean logic, probability, and Bayesian and Monte Carlo techniques. The system reliability, a numerical-valued function of the component reliabilities, is derived by the method of inclusion-exclusion, also known as Poincaré's theorem, from the "minimal paths" or the "minimal cuts". The failure-history data are "sufficient statistics" for the parameters of Bayesian conjugate prior distributions (c.p.d.'s) on the component reliabilities, "beta" for attributes and "negative-log gamma" for time-to-failure.

Functions Performed

If the system minimal success states or "minimal paths" are input, a polynomial is generated which gives the system reliability as a function of the component reliabilities. By substituting component reliabilities into this function, the system reliability is obtained. Dually, if the system minimal failure states or "minimal cuts" are input, a polynomial is generated which gives the system unreliability as a function of the component unreliabilities.

SPARCS assesses by Monte Carlo. Through this process, the system reliability estimate for each Monte Carlo trial is obtained by substituting component probabilities into the function. At each trial, for each component, a value of the reliability is generated from the c.p.d. and

substituted into the system function, to obtain a value of the system reliability for that trial. The resulting "empirical" distribution of system reliabilities, obtained over a series of trials, provides the basis for an assessment. Percentage points on that distribution are interpreted as system reliability confidence limits. The corresponding MTBF confidence limits are calculated, based on the simple relationship between the reliability and the MTBF.

FUNCTION DESCRIPTION

There are 15 specific functions performed by the computer program and they are a part of the internal documentation included in the first four and one-half pages of computer output. The functions and procedure names included in the program are as follows:

ALGAMA	:	COMPUTES LOG OF GAMMA FUNCTION.
CABTA	:	GENERATES BETA DEVIATES.
COMPUTE	:	CALCULATES MODULE OR SYSTEM RELIABILITY OR UNRELIABILITY.
CATAGEN	:	COMPUTES COMPONENT PRIOR DISTRIBUTION MEAN RELIABILITIES.
EQGEN	:	GENERATES PROBABILITY EQUATIONS.
EQPUT	:	PRINTS OUT PROBABILITY EQUATIONS.
GAUSF.	:	GENERATES (0,1) NORMAL DEVIATES.
HDLINE	:	PRINTS OUT SYSTEM IDENTIFICATION.
HISINF	:	PRINTS OUT FAILURE-HISTORY DATA FOR COMPONENTS AND PRIOR MEANS.
INPUT1,2,3,4	:	READS INPUT DATA.
MEANREL	:	COMPUTES MEAN COMPONENT RELIABILITIES FROM FAILURE-HISTORY DATA.
RANF	:	GENERATES UNIFORM PSEUDORANDOM DEVIATES.
RGAMA	:	GENERATES GAMMA DEVIATES.
SORT	:	PERFORMS 'SHELL' SORT.
STAT	:	COMPUTES AND PRINTS R AND MTBF AVERAGES, VARIANCES & PERCENTILES.

USAGE INSTRUCTIONS

Computer Input Terms

ATYPE	-	FLAG INDICATING RELIABILITY OR UNRELIABILITY COMPUTATION
MINPTH	-	BINARY VECTORS FOR MINIMAL PATH
NARG	-	SEED FOR RANDOM NUMBER GENERATION
NCOM	-	NUMBER OF COMPONENTS IN SYSTEM
NMOD	-	NUMBER OF MODULES IN SYSTEM
NPATH	-	NUMBER OF MINIMAL STATES IN SYSTEM
SIMNUM	-	NUMBER OF SIMULATIONS TO BE PERFORMED
SYSID	-	SYSTEM IDENTIFICATION
TIME	-	MISSION TIME
UNIT	-	TIME UNIT

Capacity

SPARCS-2 can process a system consisting of up to 128 modules or components in any configuration with up to 256 minimal states. Likewise, a module within the system can have 128 components and 256 minimal states. A probability equation can have up to 3500 terms.

Making Alterations to SPARCS-2 Program

Minor alterations can be made to SPARCS without great difficulty. For example, increasing the capacity for terms in the probability equation under a dynamic storage allocation can be done by augmenting the numerical argument for TERMS in the statement

```
DCL (MINPTH (128), TERMS (3500)) BIT (128) VAR.
```

If possible, alterations, when they are necessary, should be confined to

minor changes like increases in capacity or input-output modifications. Mathematical procedures such as EQGEN (probability equation generator), CABTA (beta random deviate generator), and RGAMA (gamma random deviate generator) can be changed only if there is sufficient background mathematical analysis. For example, the random deviate generators are checked out on 32-bit IBM-370 machines; extensive reprogramming would have to be done for other word sizes, or for the computer products of other manufacturers.

Substitution of Beta Components for Gamma Components

As this report went to press, it became apparent that certain field changes have to be made to the gamma generator incorporated into SPARCS-2. Until these changes are made, it will be necessary for users to substitute beta for gamma. This means treat all time-to-failure gamma components as if they were zero-one attribute beta components. This is easily done as explained below.

A gamma component has two historical data inputs: t, number of missions, and r, number of failures. A beta component also has two pieces of input data: s, successes, and r, failures. Both s and r are integers, whereas the gamma t is not necessarily an integer. All that is needed to make the substitution is an integer value of s which is a function of t and r. The following formula makes that conversion. Let

$$z = \left(\frac{t + 1}{t + 2}\right)^{r + 1}$$

denote the mean of the negative-log gamma conjugate prior distribution on the time-to-failure component reliability. Then

$$s \approx \frac{(b + 2)z - 1}{1 - z}; \tag{1}$$

that is, s is the integer closest to the right-hand side of (1). This conversion

was derived by equating the mean of the negative-log gamma distribution with the mean of the beta distribution.

Preparation of Inputs

The internal documentation of the program describes the preparation of the input data. The input data format is shown both for a system without modules and for a system with modules.

SYSTEM WITHOUT MODULES

DATA SET	NUMBER OF CARDS	COLUMN POSITION	DESCRIPTIONS
FIRST	1	1 - 80	ALPHANUMERIC SYSTEM IDENTIFICATION.
SECOND	1	1 - 4	NUMBER OF MONTE CARLO TRIALS.
		6 - 14	SEED VALUE FOR RANDOM NUMBER GENERATION (MUST BE A NON-ZERO POSITIVE INTEGER, <= 1.0 E+09). FOR EXAMPLE, 4527851 .
		16 - 20	SYSTEM MISSION TIME (OPTIONAL).
THIRD	1	1 - 3	(NUMERIC ZERO DENOTES NO MODULES).
		5 - 7	NUMBER OF COMPONENTS IN SYSTEM(N<129)
		9 - 13	NUMBER OF SYSTEM MINIMAL STATES (M<257).
		14	R FOR SYSTEM RELIABILITY, U FOR SYSTEM UNRELIABILITY.
		15 - 19	MISSION TIME UNITS, ALPHABETIC (OPTIONAL).
FOURTH	N	FREE FORMAT	ITEM1: '0'B FOR GAMMA (TIME-TO-FAILURE) OR '1'B FOR BETA (SUCCESS-FAILURE). ITEM2: NUMBER OF MISSIONS (GAMMA) OR SUCCESSES (BETA). ITEM3: NUMBER OF FAILURES.
FIFTH		FREE FORMAT	M MINIMAL STATES AS A STRING OF N-VECTORS SUCH AS '00110'B. COMPONENTS IN THE MINIMAL STATE ARE DENOTED BY 1, COMPONENTS NOT IN MINIMAL STATE BY 0.

EXAMPLE

```
COLUMN      000000000111111111122222222223333333333
POSITION    123456789012345678901234567890123456789

CARD  6     '10'B  '01'B
      5     '1'B    95.   3.
      4     '0'B    99.   1.
      3       0   2       2RHOUR
      2      100 4527851    100
      1     TWO COMPONENT - TWO MINIMAL PATH SYSTEM
```

SYSTEM WITH MODULES

DATA SET	NUMBER OF CARDS	COLUMN POSITION	DESCRIPTION
FIRST	1	1 - 80	ALPHANUMERIC SYSTEM IDENTIFICATION.
SECOND	1	1 - 4	NUMBER OF MONTE CARLO TRIALS.
		6 - 14	SEED VALUE FOR RANDOM NUMBER GENERATION (MUST BE A NON-ZERO POSITIVE INTEGER, <= 1.0 E+09). FOR EXAMPLE, 2365142 .
		16 - 20	SYSTEM MISSION TIME (OPTIONAL).
THIRD	1	1 - 3	NUMBER OF MODULES IN SYSTEM (K<129).
		5 - 7	NUMBER OF COMPONENTS IN SYSTEM.
		9 - 13	NUMBER OF MINIMAL STATES IN SYSTEM (J<257).
		14	R FOR SYSTEM RELIABILITY, U FOR SYSTEM UNRELIABILITY.
		15 - 19	MISSION TIME UNITS, ALPHABETIC (OPTIONAL).
FOURTH		FREE FORMAT	J SYSTEM MINIMAL STATES AS A STRING OF BINARY K-VECTORS, SUCH AS '010'B. MODULES IN MINIMAL STATE ARE DENOTED BY 1, MODULES NOT IN MINIMAL STATE BY 0.
FIFTH		1	R FOR MODULE RELIABILITY, U FOR MODULE UNRELIABILITY.
		2 - 7	MODULE IDENTIFICATION, ALPHANUMERIC.
		12 - 14	NUMBER OF COMPONENTS IN MODULE(N<129)
		16 - 18	NUMBER OF MINIMAL STATES IN MODULE (M<257).
SIXTH	M	FREE FORMAT	ITEM1: '0'B FOR GAMMA(TIME-TO-FAILURE) OR '1'B FOR BETA (SUCCESS - FAILURE). ITEM2: NUMBER OF MISSIONS(GAMMA) OR SUCCESSES(BETA). ITEM3: NUMBER OF FAILURES.
SEVENTH		FREE FORMAT	M MODULE MINIMAL STATES AS A STRING OF BINARY N-VECTORS SUCH AS '01110'B. COMPONENTS IN MINIMAL STATE ARE DENOTED BY 1, COMPONENTS NOT IN MINIMAL STATE BY 0.

* THE FIFTH, SIXTH, AND SEVENTH DATA SETS ARE REPEATED FOR EACH MODULE.

EXAMPLE

```
COLUMN    000000000111111111122222222223333333333
POSITION  123456789012345678901234567890123456789

CARD  12  '10'B  '01'B
      11  '1'B   90.  1.
      10  '0'B   92.  7.
       9  RB   2  2
       8  '01'B  '10'B
       7  '0'B  95.  5.
       6  '1'B  99.  1.
       5  UA   2  2
       4  '0'B   '10'B
       3    2   2     2RCYCLE
       2    20 2365142   50
       1  TWO MODULES AND TWO COMPCNENTS SYSTEM ANALYSIS
```

Results of Operation

Two cases of a sample problem will be used to demonstrate the rationale behind the problem design. The cards from the data deck are listed as well as the output format and a discussion of the output is included.

Sample Problem

The sample problem in this section is taken from a TRW internal publication [6] on software reliability. We show how it was run in different ways to illustrate some of the applications of the "super modularity" feature of SPARCS-2. The logical configuration can be either the system minimal paths (system reliability) or minimal cuts (system unreliability) with either the module minimal paths (module reliability) or minimal cuts (module unreliability). Since the output, a schedule of percentage points on an "empirical" distribution of the reliability, is

standardized to system reliability, the example also demonstrates the reproducibility of results.

The problem is illustrated by the flow diagram in Figure 1 and the corresponding network logic diagram in Figure 2, both of which are taken from [6]. It is identified in the reference as a "Triangle Type Determination Program (TTDP)" for structural-exercise test effectiveness measurement tools belonging to TRW's Product Assurance Confidence Evaluator (PACE) system.

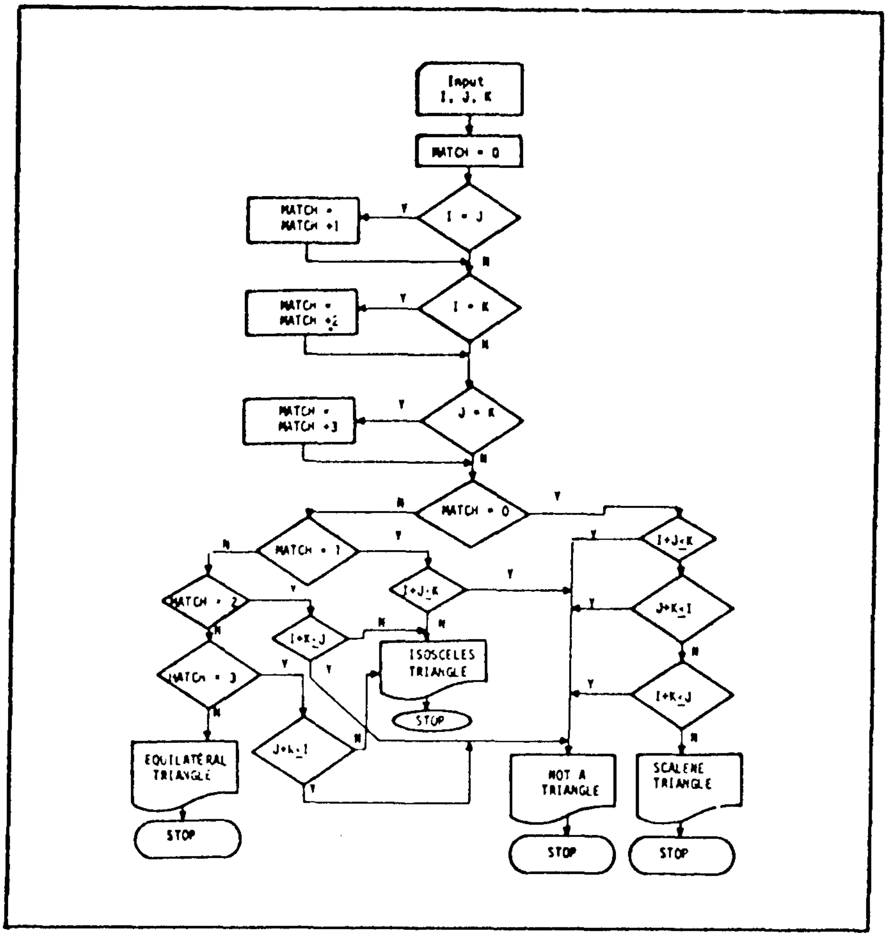

Figure 1. Flow Diagram of TTDP

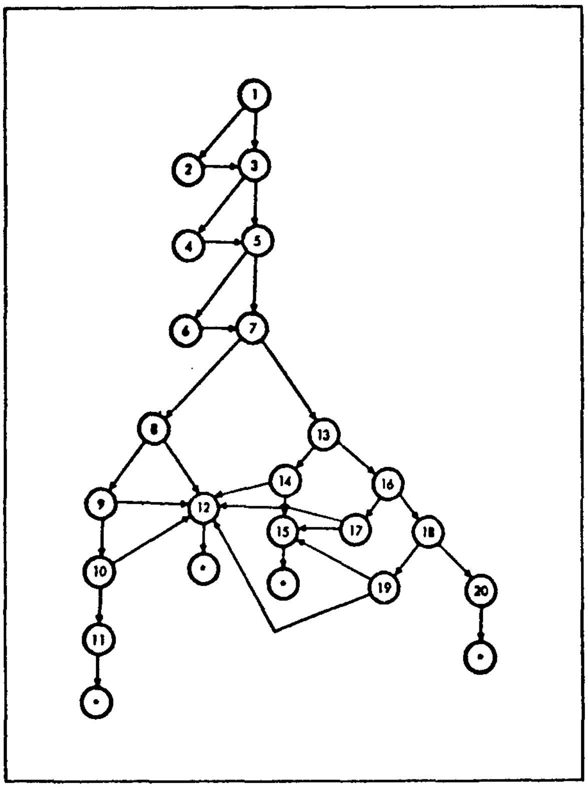

Figure 2. Network of Node-to-Node Branching Potential

The system was subdivided into two modules, A with components 1, ..., 7, and B with components 8, ..., 20. Since the SPARCS-2 program assigns sequence numbers to the components within each module by default, 8, ..., 20 were renumbered 1, ..., 13, respectively, so that there is a constant difference of 7 between the module-B component numbers in the computer output and in the description in the sequence. For example, 8 in the description is 1 in the computer output, 9 is 2, etc.

Module A represents the top half of Figure 2 and B the bottom half. Since both are needed, the system may be viewed either as two "successful" elements in series for reliability calculations, or else two failed elements

in parallel for unreliability calculations. Thus there is one minimal path

A, B

and there are two minimal cuts

A
B.

Since all four components 1, 3, 5, and 7 are needed for module A to be operative, it has one minimal path

1, 3, 5, 7

and four minimal cuts

1
3
5
7.

Components 2, 4, and 6 in module A are not essential, and do not affect the reliability calculations; module A will work even if all of them are inoperative, as long as 1, 3, 5, and 7 are all working properly.

Module B has nine minimal paths ("minimal path sets" or "tie sets")

13, 16, 18, 20
12, 13, 14
13, 14, 15
12, 13, 16, 17
13, 15, 16, 17
12, 13, 16, 18, 19
13, 15, 16, 18, 19
8, 9, 10, 11
8, 12

It is shown below that the module-B reliability equation based on these minimal paths is 91 terms long and requires a full page of computer output.

By inverting and minimalizing the minimal paths, we obtain the 26 minimal cuts ("minimal cut sets").

8, 13
8, 14, 16
9, 12, 13
10, 12, 13
11, 12, 13
8, 14, 17, 18
8, 12, 15, 16
8, 12, 15, 18
8, 12, 15, 20
9, 12, 15, 18
9, 12, 15, 20
9, 12, 14, 16
10, 12, 14, 16
10, 12, 15, 16
10, 12, 15, 18
10, 12, 15, 20
11, 12, 14, 16
11, 12, 15, 16
11, 12, 15, 18
11, 12, 15, 20
8, 14, 17, 19, 20
9, 12, 14, 17, 18
9, 12, 14, 17, 20
10, 12, 14, 17, 18
9, 12, 14, 17, 20
11, 12, 14, 17, 18

The module-B unreliability equation based on these 26 minimal cuts has 421 terms and requires four and one-half pages of computer output.

There are many different optional ways of inputting these data, based on either the paths (reliability) or the cuts (unreliability). The TRW-TTDP problem was run with eight different cases with the same historical input data for all components, and 20 Monte Carlo trials for each case.

1. System reliability - module A reliability - module B reliability.

2. System reliability - module A unreliability - module B reliability.

3. System reliability - module A unreliability - module B unreliability.

4. System unreliability - module A unreliability - module B reliability.

5. System reliability - module A unreliability - module B reliability.

6. System reliability - module A unreliability - module B unreliability.

7. System unreliability - module A reliability - module B reliability.

8. System unreliability - module A reliability - module B unreliability.

We discuss Cases 1 and 6 to illustrate the input and the output formats. The results also show surprising consistency, demonstrating that an assessment with SPARCS-2 can be accomplished with relatively few Monte Carlo trials.

Case 1 is "all reliability", reliability for the system, and for module A and module B. The cards of the input data deck are listed below.

```
                                                   80/80

       0000000000111111111122222222223333333
       1234567890123456789012345678901234567
CARD
0001        SOFTWARE RELIABILITY COMPUTATION
0002      20 15783       100
0003      2    2      1 RHOUR
0004   '11'B
0005   RA                7    1
0006          '0'B     96.5      2.
0007          '1'B     99.       1.
0008          '1'B     99.       1.
0009          '1'B     99.       1.
0010          '0'B     96.5      2.
0011          '1'B     99.       1.
0012          '1'B     99.       1.
0013   '1010101'B
0014   RB                13   9
0015          '0'B     96.5      2.
0016          '1'B     99.       1.
0017          '0'B     96.5      2.
0018          '1'B     99.       1.
0019          '0'B     96.5      2.
0020          '1'B     99.       1.
0021          '0'B     96.5      2.
0022          '1'B     99.       1.
0023          '0'B     96.5      2.
0024          '1'B     99.       1.
0025          '0'B     96.5      2.
0026          '1'B     99.       1.
0027          '0'B     96.5      2.
0028   '0000010010101'B
0029   '0000111000000'B
0030   '0000011100000'B
0031   '0000110011000'B
0032   '0000010111000'B
0033   '0000110010110'B
0034   '0000010110110'B
0035   '1111000000000'B
0036   '1000100000000'B
0037   $ENDLIST
```

The output is listed below. Following the system ID data, we have the minimal state or states, either paths or else cuts, the system probability function, either a reliability or an unreliability equation, the module-A minimal states, probability equation and component historical data, and the component-B minimal states, equation and historical data. Then follow various sets of statistical data relating to the output and finally the empirical distribution displaying the percentage points for assessment. The output format is practically the same as that in [5] for the original version of SPARCS.

```
* S P A R C S *
SIMULATICN PROGRAM FOR THE ANALYSIS OF THE RELIABILITY OF COMPLEX SYSTEMS


SYSTEM IDENTIFICATION           SOFTWARE RELIABILITY COMPUTATION - J R BROWN AND LIPOW
NUMBER OF SIMULATICNS           20
NUMBER OF MODULES               2
NUMBER OF COMPONENTS            2
NUMBER OF MINIMAL PATHS         1
TYPE OF ANALYSIS                RELIABILITY


SPARCS :: EQUATION GENERATION ROUTINE

THE    1 MINIMAL PATH   FOR SYSTEM

<A,B>

 NUMBER OF TERMS IN EQUATION:    1


  R    = R R
   SYS    A B

THE    1 MINIMAL PATH   FCR MODULE A

<1,3,5,7>

 NUMBER OF TERMS IN EQUATION:    1


  R    = R R R R
   A      1 3 5 7
```

HISTORICAL INFORMATION FOR EACH COMPONENT IN MODULE A

COMPONENT	TYPE	SUCCESS (BETA) EQUIVALENT MISSIONS(GAMMA)	FAILURES	PRIOR DISTRIBUTION MEAN
1	TIME-TO-FAILURE(GAMMA)	96.50	2.00	0.96985
2	ATTRIBUTE(BETA)	99.00	1.00	0.98039
3	ATTRIBUTE(BETA)	99.00	1.00	0.98039
4	ATTRIBUTE(BETA)	99.00	1.00	0.98039
5	TIME-TO-FAILURE(GAMMA)	96.50	2.00	0.96985
6	ATTRIBUTE(BETA)	99.00	1.00	0.98039
7	ATTRIBUTE(BETA)	99.00	1.00	0.98039

THE 9 MINIMAL PATHS FOR MODULE B

<6,9,11,13>
<5,6,7>
<6,7,8>
<5,6,9,10>
<6,8,9,10>
<5,6,9,11,12>
<6,8,9,11,12>
<1,2,3,4>
<1,5>

NUMBER OF TERMS IN EQUATION: 91

$$R_{B} = R_{6}R_{9}R_{11}R_{13} + R_{5}R_{6}R_{7} - R_{5}R_{6}R_{7}R_{9}R_{11}R_{13} + R_{6}R_{7}R_{8} - R_{6}R_{7}R_{8}R_{9}R_{11}R_{13} - R_{5}R_{6}R_{7}R_{8} + R_{5}R_{6}R_{7}R_{8}R_{9}R_{11}R_{13} + R_{5}R_{6}R_{9}R_{10} - R_{5} \ast$$

$$R_{6}R_{9}R_{10}R_{11}R_{13} - R_{5}R_{6}R_{7}R_{9}R_{10} + R_{5}R_{6}R_{7}R_{9}R_{10}R_{11}R_{13} + R_{6}R_{8}R_{9}R_{10} - R_{6}R_{8}R_{9}R_{10}R_{11}R_{13} + R_{5}R_{6}R_{7}R_{8}R_{9}R_{10} - R_{5}R_{6}R_{7}R_{8}R_{9}R_{10}R_{11}R_{13}$$

$$- R_{6}R_{7}R_{8}R_{9}R_{10} + R_{6}R_{7}R_{8}R_{9}R_{10}R_{11}R_{13} - R_{5}R_{6}R_{8}R_{9}R_{10} + R_{5}R_{6}R_{8}R_{9}R_{10}R_{11}R_{13} + R_{5}R_{6}R_{9}R_{11}R_{12} - R_{5}R_{6}R_{9}R_{11}R_{12}R_{13} - R_{5}R_{6}R_{7}R_{9}R_{11} \ast$$

$$R_{12} + R_{5}R_{6}R_{7}R_{9}R_{11}R_{12}R_{13} - R_{5}R_{6}R_{9}R_{10}R_{11}R_{12} + R_{5}R_{6}R_{9}R_{10}R_{11}R_{12}R_{13} + R_{5}R_{6}R_{7}R_{9}R_{10}R_{11}R_{12} - R_{5}R_{6}R_{7}R_{9}R_{10}R_{11}R_{12}R_{13} + R_{6}R_{8}R_{9} \ast$$

$$R_{11}R_{12} - R_{6}R_{8}R_{9}R_{11}R_{12}R_{13} + R_{5}R_{6}R_{7}R_{8}R_{9}R_{11}R_{12} - R_{5}R_{6}R_{7}R_{8}R_{9}R_{11}R_{12}R_{13} - R_{6}R_{7}R_{8}R_{9}R_{11}R_{12} + R_{6}R_{7}R_{8}R_{9}R_{11}R_{12}R_{13} + R_{5}R_{6}R_{8}R_{9} \ast$$

$$R_{10}R_{11}R_{12} - R_{5}R_{6}R_{8}R_{9}R_{10}R_{11}R_{12}R_{13} - R_{5}R_{6}R_{7}R_{8}R_{9}R_{10}R_{11}R_{12} + R_{5}R_{6}R_{7}R_{8}R_{9}R_{10}R_{11}R_{12}R_{13} - R_{6}R_{8}R_{9}R_{10}R_{11}R_{12} + R_{6}R_{8}R_{9}R_{10}R_{11}R_{12} \ast$$

$$R_{13} + R_{6}R_{7}R_{8}R_{9}R_{10}R_{11}R_{12} - R_{6}R_{7}R_{8}R_{9}R_{10}R_{11}R_{12}R_{13} - R_{5}R_{6}R_{8}R_{9}R_{11}R_{12} + R_{5}R_{6}R_{8}R_{9}R_{11}R_{12}R_{13} + R_{1}R_{2}R_{3}R_{4} - R_{1}R_{2}R_{3}R_{4}R_{6}R_{9}R_{11}R_{13}$$

$$- R_{1}R_{2}R_{3}R_{4}R_{6}R_{7}R_{8} + R_{1}R_{2}R_{3}R_{4}R_{6}R_{7}R_{8}R_{9}R_{11}R_{13} + R_{1}R_{2}R_{3}R_{4}R_{5}R_{6}R_{7}R_{8} - R_{1}R_{2}R_{3}R_{4}R_{5}R_{6}R_{7}R_{8}R_{9}R_{11}R_{13} - R_{1}R_{2}R_{3}R_{4}R_{6}R_{8}R_{9}R_{10} + R_{1} \ast$$

$$R_{2}R_{3}R_{4}R_{6}R_{8}R_{9}R_{10}R_{11}R_{13} - R_{1}R_{2}R_{3}R_{4}R_{5}R_{6}R_{7}R_{8}R_{9}R_{10} + R_{1}R_{2}R_{3}R_{4}R_{5}R_{6}R_{7}R_{8}R_{9}R_{10}R_{11}R_{13} + R_{1}R_{2}R_{3}R_{4}R_{6}R_{7}R_{8}R_{9}R_{10} - R_{1}R_{2}R_{3}R_{4}R_{6}R_{7}R_{8} \ast$$

$$R_{9}R_{10}R_{11}R_{13} + R_{1}R_{2}R_{3}R_{4}R_{5}R_{6}R_{8}R_{9}R_{10} - R_{1}R_{2}R_{3}R_{4}R_{5}R_{6}R_{8}R_{9}R_{10}R_{11}R_{13} - R_{1}R_{2}R_{3}R_{4}R_{6}R_{8}R_{9}R_{11}R_{12} + R_{1}R_{2}R_{3}R_{4}R_{6}R_{8}R_{9}R_{11}R_{12}R_{13} - R_{1} \ast$$

$$R_{2}R_{3}R_{4}R_{5}R_{6}R_{7}R_{8}R_{9}R_{11}R_{12} + R_{1}R_{2}R_{3}R_{4}R_{5}R_{6}R_{7}R_{8}R_{9}R_{11}R_{12}R_{13} + R_{1}R_{2}R_{3}R_{4}R_{6}R_{7}R_{8}R_{9}R_{11}R_{12} - R_{1}R_{2}R_{3}R_{4}R_{6}R_{7}R_{8}R_{9}R_{11}R_{12}R_{13} - R_{1}R_{2}R_{3} \ast$$

$$R_{4}R_{5}R_{6}R_{8}R_{9}R_{10}R_{11}R_{12} + R_{1}R_{2}R_{3}R_{4}R_{5}R_{6}R_{8}R_{9}R_{10}R_{11}R_{12}R_{13} + R_{1}R_{2}R_{3}R_{4}R_{5}R_{6}R_{7}R_{8}R_{9}R_{10}R_{11}R_{12} - R_{1}R_{2}R_{3}R_{4}R_{5}R_{6}R_{7}R_{8}R_{9}R_{10}R_{11}R_{12}R_{13}$$

$$+ R_{1}R_{2}R_{3}R_{4}R_{6}R_{8}R_{9}R_{10}R_{11}R_{12} - R_{1}R_{2}R_{3}R_{4}R_{6}R_{8}R_{9}R_{10}R_{11}R_{12}R_{13} - R_{1}R_{2}R_{3}R_{4}R_{6}R_{7}R_{8}R_{9}R_{10}R_{11}R_{12} + R_{1}R_{2}R_{3}R_{4}R_{6}R_{7}R_{8}R_{9}R_{10}R_{11}R_{12}R_{13}$$

$$+ R_{1}R_{2}R_{3}R_{4}R_{5}R_{6}R_{8}R_{9}R_{11}R_{12} - R_{1}R_{2}R_{3}R_{4}R_{5}R_{6}R_{8}R_{9}R_{11}R_{12}R_{13} + R_{1}R_{5} - R_{1}R_{5}R_{6}R_{9}R_{11}R_{13} - R_{1}R_{5}R_{6}R_{7} + R_{1}R_{5}R_{6}R_{7}R_{9}R_{11}R_{13} - R_{1}R_{5} \ast$$

$$R_{6}R_{9}R_{10} + R_{1}R_{5}R_{6}R_{9}R_{10}R_{11}R_{13} + R_{1}R_{5}R_{6}R_{7}R_{9}R_{10} - R_{1}R_{5}R_{6}R_{7}R_{9}R_{10}R_{11}R_{13} - R_{1}R_{5}R_{6}R_{9}R_{11}R_{12} + R_{1}R_{5}R_{6}R_{9}R_{11}R_{12}R_{13} + R_{1}R_{5}R_{6}R_{7} \ast$$

$$R_{9}R_{11}R_{12} - R_{1}R_{5}R_{6}R_{7}R_{9}R_{11}R_{12}R_{13} + R_{1}R_{5}R_{6}R_{9}R_{10}R_{11}R_{12} - R_{1}R_{5}R_{6}R_{9}R_{10}R_{11}R_{12}R_{13} - R_{1}R_{5}R_{6}R_{7}R_{9}R_{10}R_{11}R_{12} + R_{1}R_{5}R_{6}R_{7}R_{9}R_{10}R_{11} \ast$$

$$R_{12}R_{13} - R_{1}R_{2}R_{3}R_{4}R_{5} + R_{1}R_{2}R_{3}R_{4}R_{5}R_{6}R_{9}R_{11}R_{13}$$

HISTORICAL INFORMATION FOR EACH COMPONENT IN MODULE 8

COMPONENT	TYPE	SUCCESS (BETA) EQUIVALENT MISSIONS(GAMMA)	FAILURES	PRIOR DISTRIBUTION MEAN
1	TIME-TO-FAILURE(GAMMA)	96.50	2.00	0.96985
2	ATTRIBUTE(BETA)	99.00	1.00	0.98039
3	TIME-TO-FAILURE(GAMMA)	96.50	2.00	0.96985
4	ATTRIBUTE(BETA)	99.00	1.00	0.98039
5	TIME-TO-FAILURE(GAMMA)	96.50	2.00	0.96985
6	ATTRIBUTE(BETA)	99.00	1.00	0.98039
7	TIME-TO-FAILURE(GAMMA)	96.50	2.00	0.96985
8	ATTRIBUTE(BETA)	99.00	1.00	0.98039
9	TIME-TO-FAILURE(GAMMA)	96.50	2.00	0.96985
10	ATTRIBUTE(BETA)	99.00	1.00	0.98039
11	TIME-TO-FAILURE(GAMMA)	96.50	2.00	0.96985
12	ATTRIBUTE(BETA)	99.00	1.00	0.98039
13	TIME-TO-FAILURE(GAMMA)	96.50	2.00	0.96985

SPARCS :: SYSTEM SIMULATION ROUTINE

SYSTEM RELIABILITY CALCULATED FROM MEAN COMPONENT RELIABILITIES IS 0.903486; SYSTEM UNRELIABILITY IS 0.096514

AVERAGE SYSTEM RELIABILITY FROM 20 MONTE CARLO TRIALS IS 0.901238 ; AVERAGE SYSTEM UNRELIABILTY IS 0.098762

VARIANCE 0.000671

STANDARD DEVIATION 0.025896

THE MISSION TIME IS 100.000 HOUR

THE ESTIMATED SYSTEM MTBF BASED UPON MEAN COMPONENT RELIABILITIES IS 9.85000000E+02

THE ESTIMATED SYSTEM MTBF BASED UPON MEAN SYSTEM RELIABILITY 20 MONTE CARLO TRIALS IS 9.61669678E+02

PERCENTILE	RELIABILITY PERCENTILE POINTS	MTBF PERCENTILE POINTS	
5.0 PERCENT	0.848968	6.10748047E+02	HOUR
10.0 PERCENT	0.865494	6.92260010E+02	HOUR
20.0 PERCENT	0.878241	7.70209473E+02	HOUR
25.0 PERCENT	0.884024	8.11221191E+02	HOUR
50.0 PERCENT	0.904296	9.94052002E+02	HOUR
75.0 PERCENT	0.921494	1.22310010E+03	HOUR
80.0 PERCENT	0.924841	1.27986133E+03	HOUR
90.0 PERCENT	C.929575	1.36934155E+03	HOUR
95.0 PERCENT	0.933471	1.45253003E+03	HOUR
97.5 PERCENT	0.937314	1.54470239E+03	HOUR
99.0 PERCENT	C.939619	1.60563623E+03	HOUR

Case 6 follows. Note that the module-B five-page probability equation with 421 terms based on 26 cuts is given with U's (for "unreliability") rather than R's (for "reliability"), as in the reliability equation for Case 1.

```
       000000000111111111122222222223333333333444444444455555555556666666666777777
       123456789012345678901234567890123456789012345678901234567890123456789012345
CARD
           SOFTWARE RELIABILITY - UNRELIABILITY COMPUTATION - J R BROWN AND LIPOW
         20 2794360    100
         2    2       1 RHOUR
       '11'B
       UA                7    4
            '0'B     96.5     2.
            '1'B     99.      1.
            '1'B     99.      1.
            '1'B     99.      1.
            '0'B     96.5     2.
            '1'B     99.      1.
            '1'B     99.      1.
       '1000000'B
       '0010000'B
       '0000100'B
       '0000001'B
       UB               13   26
            '0'B     96.5     2.
            '1'B     99.      1.
            '0'B     96.5     2.
            '1'B     99.      1.
            '0'B     96.5     2.
            '1'B     99.      1.
            '0'B     96.5     2.
            '1'B     99.      1.
            '0'B     96.5     2.
            '1'B     99.      1.
            '0'B     96.5     2.
            '1'B     99.      1.
            '0'B     96.5     2.
       '1000010000000'B
       '1000001010000'B
       '0100110000000'B
       '0010110000000'B
       '0001110000000'B
       '1000001001100'B
       '1000100110000'B
       '1000100100100'B
       '1000100100001'B
       '0100100100100'B
       '0100100100001'B
       '0100101010000'B
       '0010101010000'B
       '0010100110000'B
       '0010100100100'B
       '0010100100001'B
       '0001101010000'B
       '0001100110000'B
       '0001100100100'B
       '0001100100001'B
       '1000001001011'B
       '0100101001100'B
       '0100101001001'B
       '0010101001100'B
       '0100101001001'B
       '0001101001100'B
       $ENDLIST
```

* S P A R C S *
SIMULATION PROGRAM FOR THE ANALYSIS OF THE RELIABILITY OF COMPLEX SYSTEMS

SYSTEM IDENTIFICATION	SOFTWARE RELIABILITY - UNRELIABILITY COMPUTATION - J R BROWN AND LIPOW
NUMBER OF SIMULATIONS	20
NUMBER OF MODULES	2
NUMBER OF COMPONENTS	2
NUMBER OF MINIMAL PATHS	1
TYPE OF ANALYSIS	RELIABILITY

SPARCS :: EQUATION GENERATION ROUTINE

THE 1 MINIMAL PATH FOR SYSTEM

<A.B>

NUMBER OF TERMS IN EQUATION: 1

$$R_{SYS} = R_A R_B$$

THE 4 MINIMAL CUTS FOR MODULE A

<1>
<3>
<5>
<7>

NUMBER OF TERMS IN EQUATION: 15

$$U_A = U_1 + U_3 - U_1U_3 + U_5 - U_1U_5 - U_3U_5 + U_1U_3U_5 + U_7 - U_1U_7 - U_3U_7 + U_1U_3U_7 - U_5U_7 + U_1U_5U_7 + U_3U_5U_7 - U_1U_3U_5U_7$$

HISTORICAL INFORMATION FOR EACH COMPONENT IN MODULE A

COMPONENT	TYPE	SUCCESS (BETA) EQUIVALENT MISSIONS(GAMMA)	FAILURES	PRIOR DISTRIBUTION MEAN
1	TIME-TO-FAILURE(GAMMA)	96.50	2.00	0.969851
2	ATTRIBUTE(BETA)	99.00	1.00	0.980392
3	ATTRIBUTE(BETA)	99.00	1.00	0.980392
4	ATTRIBUTE(BETA)	99.00	1.00	0.980392
5	TIME-TO-FAILURE(GAMMA)	96.50	2.00	0.969851
6	ATTRIBUTE(BETA)	99.00	1.00	0.980392
7	ATTRIBUTE(BETA)	99.00	1.00	0.980392

THE 26 MINIMAL CUTS FOR MODULE B

<1,6>
<1,7,9>
<2,5,6>
<3,5,6>
<4,5,6>
<1,7,10,11>
<1,5,8,9>
<1,5,8,11>
<1,5,8,13>
<2,5,8,11>
<2,5,8,13>
<2,5,7,9>
<3,5,7,9>
<3,5,8,9>
<3,5,8,11>
<3,5,8,13>
<4,5,7,9>
<4,5,8,9>
<4,5,8,11>
<4,5,8,13>
<1,7,10,12,13>
<2,5,7,10,11>
<2,5,7,10,13>
<3,5,7,10,11>
<2,5,7,10,13>
<4,5,7,10,11>

NUMBER OF TERMS IN EQUATION: 421

$$U_8 = U_1U_6 + U_1U_7U_9 - U_1U_6U_7U_9 + U_2U_5U_6 - U_1U_2U_5U_6 + U_3U_5U_6 - U_1U_3U_5U_6 - U_2U_3U_5U_6 + U_1U_2U_3U_5U_6 + U_4U_5U_6 - U_1U_4U_5U_6 \ast$$

$$- U_2U_4U_5U_6 + U_1U_2U_4U_5U_6 - U_3U_4U_5U_6 + U_1U_3U_4U_5U_6 + U_2U_3U_4U_5U_6 - U_1U_2U_3U_4U_5U_6 + U_1U_7U_{10}U_{11} - U_1U_6U_7U_{10}U_{11} - U_1U_7U_9 \ast$$

$$U_{10}U_{11} + U_1U_6U_7U_9U_{10}U_{11} + U_1U_5U_8U_9 - U_1U_5U_6U_8U_9 - U_1U_5U_7U_8U_9 + U_1U_5U_6U_7U_8U_9 + U_1U_5U_8U_{11} - U_1U_5U_6U_8U_{11} - U_1U_5U_7U_8U_{10} \ast$$

$$U_{11} + U_1U_5U_6U_7U_8U_{10}U_{11} + U_1U_5U_7U_8U_9U_{10}U_{11} - U_1U_5U_6U_7U_8U_9U_{10}U_{11} - U_1U_5U_8U_9U_{11} + U_1U_5U_6U_8U_9U_{11} + U_1U_5U_8U_{13} - U_1U_5U_6 \ast$$

$$U_8U_{13} - U_1U_5U_8U_9U_{13} + U_1U_5U_6U_8U_9U_{13} - U_1U_5U_8U_{11}U_{13} + U_1U_5U_6U_8U_{11}U_{13} + U_1U_5U_8U_9U_{11}U_{13} - U_1U_5U_6U_8U_9U_{11}U_{13} + U_2U_5U_8U_{11} \ast$$

$$+ U_1U_2U_5U_6U_8U_{11} - U_2U_5U_6U_8U_{11} - U_1U_2U_5U_8U_{11} + U_2U_5U_8U_{13} + U_1U_2U_5U_6U_8U_{13} - U_2U_5U_6U_8U_{13} + U_1U_2U_5U_8U_{11}U_{13} - U_1U_2U_5U_6 \ast$$

$$U_8U_{11}U_{13} - U_1U_2U_5U_8U_{13} - U_2U_5U_8U_{11}U_{13} + U_2U_5U_6U_8U_{11}U_{13} + U_2U_5U_7U_9 + U_1U_2U_5U_6U_7U_9 - U_1U_2U_5U_7U_9 - U_2U_5U_6U_7U_9 + U_1U_2 \ast$$

$$U_5U_7U_8U_9U_{11} - U_1U_2U_5U_6U_7U_8U_9U_{11} + U_1U_2U_5U_7U_8U_9U_{13} - U_1U_2U_5U_6U_7U_8U_9U_{13} - U_1U_2U_5U_7U_8U_9U_{11}U_{13} + U_1U_2U_5U_6U_7U_8U_9U_{11}U_{13}$$

$$- U_2U_5U_7U_8U_9U_{11} + U_2U_5U_6U_7U_8U_9U_{11} - U_2U_5U_7U_8U_9U_{13} + U_2U_5U_6U_7U_8U_9U_{13} + U_2U_5U_7U_8U_9U_{11}U_{13} - U_2U_5U_6U_7U_8U_9U_{11}U_{13} + U_3 \ast$$

$$U_5U_7U_9 + U_1U_3U_5U_6U_7U_9 - U_1U_3U_5U_7U_9 + U_2U_3U_5U_6U_7U_9 - U_1U_2U_3U_5U_6U_7U_9 - U_3U_5U_6U_7U_9 - U_2U_3U_5U_7U_9 + U_1U_2U_3U_5U_7U_9 + U_3U_5 \ast$$

$U_{8}U_{9} + U_{1}U_{3}U_{5}U_{6}U_{8}U_{9} + U_{1}U_{3}U_{5}U_{7}U_{8}U_{9} - U_{1}U_{3}U_{5}U_{6}U_{7}U_{8}U_{9} - U_{3}U_{5}U_{6}U_{8}U_{9} - U_{1}U_{3}U_{5}U_{8}U_{9} - U_{1}U_{2}U_{3}U_{5}U_{7}U_{8}U_{9}U_{11} + U_{1}U_{2}U_{3}U_{5}U_{6}U_{7}U_{8}$ *

$U_{9}U_{11} - U_{1}U_{2}U_{3}U_{5}U_{7}U_{8}U_{9}U_{13} + U_{1}U_{2}U_{3}U_{5}U_{6}U_{7}U_{8}U_{9}U_{13} + U_{1}U_{2}U_{3}U_{5}U_{7}U_{8}U_{9}U_{11}U_{13} - U_{1}U_{2}U_{3}U_{5}U_{6}U_{7}U_{8}U_{9}U_{11}U_{13} + U_{2}U_{3}U_{5}U_{7}U_{8}U_{9}U_{11}$ *

$- U_{2}U_{3}U_{5}U_{6}U_{7}U_{8}U_{9}U_{11} + U_{2}U_{3}U_{5}U_{7}U_{8}U_{9}U_{13} - U_{2}U_{3}U_{5}U_{6}U_{7}U_{8}U_{9}U_{13} - U_{2}U_{3}U_{5}U_{7}U_{8}U_{9}U_{11}U_{13} + U_{2}U_{3}U_{5}U_{6}U_{7}U_{8}U_{9}U_{11}U_{13} - U_{3}U_{5}U_{7}U_{8}$ *

$U_{9} + U_{3}U_{5}U_{6}U_{7}U_{8}U_{9} + U_{3}U_{5}U_{8}U_{11} + U_{1}U_{3}U_{5}U_{6}U_{8}U_{11} + U_{2}U_{3}U_{5}U_{6}U_{8}U_{11} - U_{1}U_{2}U_{3}U_{5}U_{6}U_{8}U_{11} - U_{3}U_{5}U_{6}U_{8}U_{11} + U_{1}U_{3}U_{5}U_{8}U_{9}U_{11} - U_{1}$ *

$U_{3}U_{5}U_{6}U_{8}U_{9}U_{11} - U_{1}U_{3}U_{5}U_{8}U_{11} - U_{2}U_{3}U_{5}U_{8}U_{11} + U_{1}U_{2}U_{3}U_{5}U_{8}U_{11} - U_{3}U_{5}U_{8}U_{9}U_{11} + U_{3}U_{5}U_{6}U_{8}U_{9}U_{11} + U_{3}U_{5}U_{8}U_{13} + U_{1}U_{3}U_{5}U_{6}U_{8}U_{13}$

$+ U_{2}U_{3}U_{5}U_{6}U_{8}U_{13} - U_{1}U_{2}U_{3}U_{5}U_{6}U_{8}U_{13} - U_{3}U_{5}U_{6}U_{8}U_{13} + U_{1}U_{3}U_{5}U_{8}U_{9}U_{13} - U_{1}U_{3}U_{5}U_{6}U_{8}U_{9}U_{13} + U_{1}U_{3}U_{5}U_{8}U_{11}U_{13} - U_{1}U_{3}U_{5}U_{6}U_{8}U_{11}$ *

$U_{13} - U_{1}U_{3}U_{5}U_{8}U_{9}U_{11}U_{13} + U_{1}U_{3}U_{5}U_{6}U_{8}U_{9}U_{11}U_{13} - U_{1}U_{3}U_{5}U_{8}U_{13} + U_{2}U_{3}U_{5}U_{8}U_{11}U_{13} + U_{1}U_{2}U_{3}U_{5}U_{6}U_{8}U_{11}U_{13} - U_{2}U_{3}U_{5}U_{6}U_{8}U_{11}U_{13}$

$- U_{1}U_{2}U_{3}U_{5}U_{8}U_{11}U_{13} - U_{2}U_{3}U_{5}U_{8}U_{13} + U_{1}U_{2}U_{3}U_{5}U_{8}U_{13} - U_{3}U_{5}U_{8}U_{9}U_{13} + U_{3}U_{5}U_{6}U_{8}U_{9}U_{13} - U_{3}U_{5}U_{8}U_{11}U_{13} + U_{3}U_{5}U_{6}U_{8}U_{11}U_{13} + U_{3}$ *

$U_{5}U_{8}U_{9}U_{11}U_{13} - U_{3}U_{5}U_{6}U_{8}U_{9}U_{11}U_{13} + U_{4}U_{5}U_{7}U_{9} + U_{1}U_{4}U_{5}U_{6}U_{7}U_{9} - U_{1}U_{4}U_{5}U_{7}U_{9} + U_{2}U_{4}U_{5}U_{6}U_{7}U_{9} - U_{1}U_{2}U_{4}U_{5}U_{6}U_{7}U_{9} + U_{3}U_{4}U_{5}U_{6}$ *

$U_{7}U_{9} - U_{1}U_{3}U_{4}U_{5}U_{6}U_{7}U_{9} - U_{2}U_{3}U_{4}U_{5}U_{6}U_{7}U_{9} + U_{1}U_{2}U_{3}U_{4}U_{5}U_{6}U_{7}U_{9} - U_{4}U_{5}U_{6}U_{7}U_{9} - U_{2}U_{4}U_{5}U_{7}U_{9} + U_{1}U_{2}U_{4}U_{5}U_{7}U_{9} - U_{3}U_{4}U_{5}U_{7}U_{9} + U_{1}$ *

$U_{3}U_{4}U_{5}U_{7}U_{9} + U_{2}U_{3}U_{4}U_{5}U_{7}U_{9} - U_{1}U_{2}U_{3}U_{4}U_{5}U_{7}U_{9} + U_{4}U_{5}U_{8}U_{9} + U_{1}U_{4}U_{5}U_{6}U_{8}U_{9} + U_{1}U_{4}U_{5}U_{7}U_{8}U_{9} - U_{1}U_{4}U_{5}U_{6}U_{7}U_{8}U_{9} + U_{3}U_{4}U_{5}U_{6}U_{8}$ *

$U_{9} - U_{1}U_{3}U_{4}U_{5}U_{6}U_{8}U_{9} - U_{4}U_{5}U_{6}U_{8}U_{9} - U_{1}U_{4}U_{5}U_{8}U_{9} - U_{1}U_{2}U_{4}U_{5}U_{7}U_{8}U_{9}U_{11} + U_{1}U_{2}U_{4}U_{5}U_{6}U_{7}U_{8}U_{9}U_{11} - U_{1}U_{2}U_{4}U_{5}U_{7}U_{8}U_{9}U_{13} + U_{1}U_{2}$ *

$U_{4}U_{5}U_{6}U_{7}U_{8}U_{9}U_{13} + U_{1}U_{2}U_{4}U_{5}U_{7}U_{8}U_{9}U_{11}U_{13} - U_{1}U_{2}U_{4}U_{5}U_{6}U_{7}U_{8}U_{9}U_{11}U_{13} + U_{2}U_{4}U_{5}U_{7}U_{8}U_{9}U_{11} - U_{2}U_{4}U_{5}U_{6}U_{7}U_{8}U_{9}U_{11} + U_{2}U_{4}U_{5}U_{7}U_{8}$ *

$U_{9}U_{13} - U_{2}U_{4}U_{5}U_{6}U_{7}U_{8}U_{9}U_{13} - U_{2}U_{4}U_{5}U_{7}U_{8}U_{9}U_{11}U_{13} + U_{2}U_{4}U_{5}U_{6}U_{7}U_{8}U_{9}U_{11}U_{13} + U_{3}U_{4}U_{5}U_{7}U_{8}U_{9} + U_{1}U_{3}U_{4}U_{5}U_{6}U_{7}U_{8}U_{9} - U_{1}U_{3}U_{4}U_{5}$ *

$U_{7}U_{8}U_{9} - U_{3}U_{4}U_{5}U_{6}U_{7}U_{8}U_{9} - U_{3}U_{4}U_{5}U_{8}U_{9} + U_{1}U_{3}U_{4}U_{5}U_{8}U_{9} + U_{1}U_{2}U_{3}U_{4}U_{5}U_{7}U_{8}U_{9}U_{11} - U_{1}U_{2}U_{3}U_{4}U_{5}U_{6}U_{7}U_{8}U_{9}U_{11} + U_{1}U_{2}U_{3}U_{4}U_{5}U_{7}U_{8}$ *

$U_{9}U_{13} - U_{1}U_{2}U_{3}U_{4}U_{5}U_{6}U_{7}U_{8}U_{9}U_{13} - U_{1}U_{2}U_{3}U_{4}U_{5}U_{7}U_{8}U_{9}U_{11}U_{13} + U_{1}U_{2}U_{3}U_{4}U_{5}U_{6}U_{7}U_{8}U_{9}U_{11}U_{13} - U_{2}U_{3}U_{4}U_{5}U_{7}U_{8}U_{9}U_{11} + U_{2}U_{3}U_{4}U_{5}U_{5}$ *

$U_{7}U_{8}U_{9}U_{11} - U_{2}U_{3}U_{4}U_{5}U_{7}U_{8}U_{9}U_{13} + U_{2}U_{3}U_{4}U_{5}U_{6}U_{7}U_{8}U_{9}U_{13} + U_{2}U_{3}U_{4}U_{5}U_{7}U_{8}U_{9}U_{11}U_{13} - U_{2}U_{3}U_{4}U_{5}U_{6}U_{7}U_{8}U_{9}U_{11}U_{13} - U_{4}U_{5}U_{7}U_{8}U_{9} + U_{4}$ *

$U_{5}U_{6}U_{7}U_{8}U_{9} + U_{4}U_{5}U_{8}U_{11} + U_{1}U_{4}U_{5}U_{6}U_{8}U_{11} + U_{2}U_{4}U_{5}U_{6}U_{8}U_{11} - U_{1}U_{2}U_{4}U_{5}U_{6}U_{8}U_{11} + U_{3}U_{4}U_{5}U_{6}U_{8}U_{11} - U_{1}U_{3}U_{4}U_{5}U_{6}U_{8}U_{11} - U_{2}U_{3}$ *

$U_{4}U_{5}U_{6}U_{8}U_{11} + U_{1}U_{2}U_{3}U_{4}U_{5}U_{6}U_{8}U_{11} - U_{4}U_{5}U_{6}U_{8}U_{11} + U_{1}U_{4}U_{5}U_{8}U_{9}U_{11} - U_{4}U_{5}U_{6}U_{8}U_{9}U_{11} - U_{1}U_{4}U_{5}U_{8}U_{11} - U_{2}U_{4}U_{5}U_{8}U_{11} + U_{1}U_{2}$ *

$U_{4}U_{5}U_{8}U_{11} + U_{3}U_{4}U_{5}U_{8}U_{9}U_{11} + U_{1}U_{3}U_{4}U_{5}U_{6}U_{8}U_{9}U_{11} - U_{3}U_{4}U_{5}U_{6}U_{8}U_{9}U_{11} - U_{1}U_{3}U_{4}U_{5}U_{8}U_{9}U_{11} - U_{3}U_{4}U_{5}U_{8}U_{11} + U_{1}U_{3}U_{4}U_{5}U_{8}U_{11} + U_{2}$ *

$U_3U_4U_5U_8U_{11} - U_1U_2U_3U_4U_5U_8U_{11} - U_4U_5U_8U_9U_{11} + U_4U_5U_6U_8U_9U_{11} + U_4U_5U_8U_{13} + U_1U_4U_5U_6U_8U_{13} + U_2U_4U_5U_6U_8U_{13} - U_1U_2U_4U_5$ *

$U_6U_8U_{13} + U_3U_4U_5U_6U_8U_{13} - U_1U_3U_4U_5U_6U_8U_{13} - U_2U_3U_4U_5U_6U_8U_{13} + U_1U_2U_3U_4U_5U_6U_8U_{13} - U_4U_5U_6U_8U_{13} + U_1U_4U_5U_8U_9U_{13} - U_1$ *

$U_4U_5U_6U_8U_9U_{13} + U_1U_4U_5U_8U_{11}U_{13} - U_1U_4U_5U_6U_8U_{11}U_{13} - U_1U_4U_5U_8U_9U_{11}U_{13} + U_1U_4U_5U_6U_8U_9U_{11}U_{13} - U_1U_4U_5U_8U_{13} + U_2U_4U_5U_8$ *

$U_{11}U_{13} + U_1U_2U_4U_5U_6U_8U_{11}U_{13} - U_2U_4U_5U_6U_8U_{11}U_{13} - U_1U_2U_4U_5U_8U_{11}U_{13} - U_2U_4U_5U_8U_{13} + U_1U_2U_4U_5U_8U_{13} + U_3U_4U_5U_8U_9U_{13} + U_1$ *

$U_3U_4U_5U_6U_8U_9U_{13} - U_3U_4U_5U_6U_8U_9U_{13} - U_1U_3U_4U_5U_8U_9U_{13} + U_3U_4U_5U_8U_{11}U_{13} + U_1U_3U_4U_5U_6U_8U_{11}U_{13} + U_2U_3U_4U_5U_6U_8U_{11}U_{13} - U_1$ *

$U_2U_3U_4U_5U_6U_8U_{11}U_{13} - U_3U_4U_5U_6U_8U_{11}U_{13} + U_1U_3U_4U_5U_8U_9U_{11}U_{13} - U_1U_3U_4U_5U_6U_8U_9U_{11}U_{13} - U_1U_3U_4U_5U_8U_{11}U_{13} - U_2U_3U_4U_5U_8$ *

$U_{11}U_{13} + U_1U_2U_3U_4U_5U_8U_{11}U_{13} - U_3U_4U_5U_8U_9U_{11}U_{13} + U_3U_4U_5U_6U_8U_9U_{11}U_{13} - U_3U_4U_5U_8U_{13} + U_1U_3U_4U_5U_8U_{13} + U_2U_3U_4U_5U_8U_{13}$

$- U_1U_2U_3U_4U_5U_8U_{13} - U_4U_5U_8U_9U_{13} + U_4U_5U_6U_8U_9U_{13} - U_4U_5U_8U_{11}U_{13} + U_4U_5U_6U_8U_{11}U_{13} + U_4U_5U_8U_9U_{11}U_{13} - U_4U_5U_6U_8U_9U_{11}$ *

$U_{13} + U_1U_7U_{10}U_{12}U_{13} - U_1U_6U_7U_{10}U_{12}U_{13} - U_1U_7U_9U_{10}U_{12}U_{13} + U_1U_6U_7U_9U_{10}U_{12}U_{13} - U_1U_7U_{10}U_{11}U_{12}U_{13} + U_1U_6U_7U_{10}U_{11}U_{12}U_{13}$

$+ U_1U_7U_9U_{10}U_{11}U_{12}U_{13} - U_1U_6U_7U_9U_{10}U_{11}U_{12}U_{13} + U_1U_5U_7U_8U_9U_{10}U_{12}U_{13} - U_1U_5U_6U_7U_8U_9U_{10}U_{12}U_{13} + U_1U_5U_7U_8U_{10}U_{11}U_{12}U_{13}$

$- U_1U_5U_6U_7U_8U_{10}U_{11}U_{12}U_{13} - U_1U_5U_7U_8U_9U_{10}U_{11}U_{12}U_{13} + U_1U_5U_6U_7U_8U_9U_{10}U_{11}U_{12}U_{13} - U_1U_5U_7U_8U_{10}U_{12}U_{13} + U_1U_5U_6U_7U_8U_{10}$ *

$U_{12}U_{13} + U_2U_5U_7U_{10}U_{11} + U_1U_2U_5U_6U_7U_{10}U_{11} + U_1U_2U_5U_7U_9U_{10}U_{11} - U_1U_2U_5U_6U_7U_9U_{10}U_{11} - U_2U_5U_6U_7U_{10}U_{11} - U_1U_2U_5U_7U_{10}U_{11}$ *

$- U_1U_2U_5U_7U_8U_9U_{10}U_{11} + U_1U_2U_5U_6U_7U_8U_9U_{10}U_{11} + U_1U_2U_5U_7U_8U_{10}U_{11} - U_1U_2U_5U_6U_7U_8U_{10}U_{11} - U_2U_5U_7U_8U_{10}U_{11} + U_2U_5U_6U_7U_8$ *

$U_{10}U_{11} - U_2U_5U_7U_9U_{10}U_{11} + U_2U_5U_6U_7U_9U_{10}U_{11} + U_2U_5U_7U_8U_9U_{10}U_{11} - U_2U_5U_6U_7U_8U_9U_{10}U_{11} - U_2U_5U_6U_7U_{10}U_{13} + U_2U_5U_7U_8U_{10}$ *

$U_{11}U_{13} - U_2U_5U_6U_7U_8U_{10}U_{11}U_{13} - U_2U_5U_7U_8U_{10}U_{13} + U_2U_5U_6U_7U_8U_{10}U_{13} - U_2U_5U_7U_9U_{10}U_{13} + U_2U_5U_6U_7U_9U_{10}U_{13} - U_2U_5U_7U_8U_9$ *

$U_{10}U_{11}U_{13} + U_2U_5U_6U_7U_8U_9U_{10}U_{11}U_{13} + U_2U_5U_7U_8U_9U_{10}U_{13} - U_2U_5U_6U_7U_8U_9U_{10}U_{13} - U_1U_2U_5U_7U_{10}U_{12}U_{13} + U_1U_2U_5U_6U_7U_{10}U_{12}U_{13}$

$+ U_1U_2U_5U_7U_9U_{10}U_{12}U_{13} - U_1U_2U_5U_6U_7U_9U_{10}U_{12}U_{13} + U_1U_2U_5U_7U_{10}U_{11}U_{12}U_{13} - U_1U_2U_5U_6U_7U_{10}U_{11}U_{12}U_{13} - U_1U_2U_5U_7U_9U_{10}U_{11}$ *

$U_{12}U_{13} + U_1U_2U_5U_6U_7U_9U_{10}U_{11}U_{12}U_{13} - U_1U_2U_5U_7U_8U_9U_{10}U_{12}U_{13} + U_1U_2U_5U_6U_7U_8U_9U_{10}U_{12}U_{13} - U_1U_2U_5U_7U_8U_{10}U_{11}U_{12}U_{13} + U_1$ *

$U_2U_5U_6U_7U_8U_{10}U_{11}U_{12}U_{13} + U_1U_2U_5U_7U_8U_9U_{10}U_{11}U_{12}U_{13} - U_1U_2U_5U_6U_7U_8U_9U_{10}U_{11}U_{12}U_{13} + U_1U_2U_5U_7U_8U_{10}U_{12}U_{13} - U_1U_2U_5U_6U_7$ *

$U_8U_{10}U_{12}U_{13} - U_2U_5U_7U_{10}U_{11}U_{13} + U_2U_5U_6U_7U_{10}U_{11}U_{13} + U_2U_5U_7U_9U_{10}U_{11}U_{13} - U_2U_5U_6U_7U_9U_{10}U_{11}U_{13} + U_3U_5U_7U_{10}U_{11} + U_1U_3$ *

$$U_5 U_6 U_7 U_{10} U_{11} + U_1 U_3 U_5 U_7 U_9 U_{10} U_{11} - U_1 U_3 U_5 U_6 U_7 U_9 U_{10} U_{11} + U_2 U_3 U_5 U_6 U_7 U_{10} U_{11} - U_1 U_2 U_3 U_5 U_6 U_7 U_{10} U_{11} - U_3 U_5 U_6 U_7 U_{10} U_{11} - U_1 *$$

$$U_3 U_5 U_7 U_{10} U_{11} - U_1 U_3 U_5 U_7 U_8 U_9 U_{10} U_{11} + U_1 U_3 U_5 U_6 U_7 U_8 U_9 U_{10} U_{11} + U_1 U_3 U_5 U_7 U_8 U_{10} U_{11} - U_1 U_3 U_5 U_6 U_7 U_8 U_{10} U_{11} + U_2 U_3 U_5 U_7 U_8 U_{10} U_{11} *$$

$$+ U_1 U_2 U_3 U_5 U_6 U_7 U_8 U_{10} U_{11} - U_2 U_3 U_5 U_6 U_7 U_8 U_{10} U_{11} - U_1 U_2 U_3 U_5 U_7 U_8 U_{10} U_{11} + U_2 U_3 U_5 U_7 U_9 U_{10} U_{11} + U_1 U_2 U_3 U_5 U_6 U_7 U_9 U_{10} U_{11} - U_1 U_2 *$$

$$U_3 U_5 U_7 U_9 U_{10} U_{11} - U_2 U_3 U_5 U_6 U_7 U_9 U_{10} U_{11} + U_1 U_2 U_3 U_5 U_7 U_8 U_9 U_{10} U_{11} - U_1 U_2 U_3 U_5 U_5 U_7 U_8 U_9 U_{10} U_{11} - U_2 U_3 U_5 U_7 U_8 U_9 U_{10} U_{11} + U_2 U_3 U_5 *$$

$$U_6 U_7 U_8 U_9 U_{10} U_{11} - U_3 U_5 U_7 U_9 U_{10} U_{11} + U_3 U_5 U_6 U_7 U_9 U_{10} U_{11} + U_3 U_5 U_7 U_8 U_9 U_{10} U_{11} - U_3 U_5 U_6 U_7 U_8 U_9 U_{10} U_{11} - U_3 U_5 U_7 U_8 U_{10} U_{11} + U_3 U_5 *$$

$$U_6 U_7 U_8 U_{10} U_{11} - U_2 U_3 U_5 U_7 U_{10} U_{11} + U_1 U_2 U_3 U_5 U_7 U_{10} U_{11} + U_2 U_5 U_7 U_{10} U_{13} + U_4 U_5 U_7 U_{10} U_{11} + U_1 U_4 U_5 U_6 U_7 U_{10} U_{11} + U_1 U_4 U_5 U_7 U_9 U_{10} *$$

$$U_{11} - U_1 U_4 U_5 U_6 U_7 U_9 U_{10} U_{11} + U_2 U_4 U_5 U_6 U_7 U_{10} U_{11} - U_1 U_2 U_4 U_5 U_6 U_7 U_{10} U_{11} + U_3 U_4 U_5 U_6 U_7 U_{10} U_{11} - U_1 U_3 U_4 U_5 U_6 U_7 U_{10} U_{11} - U_2 U_3 U_4 *$$

$$U_5 U_6 U_7 U_{10} U_{11} + U_1 U_2 U_3 U_4 U_5 U_6 U_7 U_{10} U_{11} - U_4 U_5 U_6 U_7 U_{10} U_{11} - U_1 U_4 U_5 U_7 U_{10} U_{11} - U_1 U_4 U_5 U_7 U_8 U_9 U_{10} U_{11} + U_1 U_4 U_5 U_6 U_7 U_8 U_9 U_{10} U_{11} *$$

$$+ U_1 U_4 U_5 U_7 U_8 U_{10} U_{11} - U_1 U_4 U_5 U_6 U_7 U_8 U_{10} U_{11} + U_2 U_4 U_5 U_7 U_8 U_{10} U_{11} + U_1 U_2 U_4 U_5 U_6 U_7 U_8 U_{10} U_{11} - U_2 U_4 U_5 U_6 U_7 U_8 U_{10} U_{11} - U_1 U_2 U_4 U_5 *$$

$$U_7 U_8 U_{10} U_{11} + U_2 U_4 U_5 U_7 U_9 U_{10} U_{11} + U_1 U_2 U_4 U_5 U_6 U_7 U_9 U_{10} U_{11} - U_1 U_2 U_4 U_5 U_7 U_9 U_{10} U_{11} - U_2 U_4 U_5 U_6 U_7 U_9 U_{10} U_{11} + U_1 U_2 U_4 U_5 U_7 U_8 U_9 U_{10} *$$

$$U_{11} - U_1 U_2 U_4 U_5 U_6 U_7 U_8 U_9 U_{10} U_{11} - U_2 U_4 U_5 U_7 U_8 U_9 U_{10} U_{11} + U_2 U_4 U_5 U_6 U_7 U_8 U_9 U_{10} U_{11} + U_3 U_4 U_5 U_7 U_9 U_{10} U_{11} + U_1 U_3 U_4 U_5 U_6 U_7 U_9 U_{10} U_{11} *$$

$$- U_1 U_3 U_4 U_5 U_7 U_9 U_{10} U_{11} + U_2 U_3 U_4 U_5 U_6 U_7 U_9 U_{10} U_{11} - U_1 U_2 U_3 U_4 U_5 U_6 U_7 U_9 U_{10} U_{11} - U_3 U_4 U_5 U_6 U_7 U_9 U_{10} U_{11} - U_2 U_3 U_4 U_5 U_7 U_9 U_{10} U_{11} + U_1 *$$

$$U_2 U_3 U_4 U_5 U_7 U_9 U_{10} U_{11} - U_3 U_4 U_5 U_7 U_8 U_9 U_{10} U_{11} - U_1 U_3 U_4 U_5 U_6 U_7 U_8 U_9 U_{10} U_{11} + U_1 U_3 U_4 U_5 U_7 U_8 U_9 U_{10} U_{11} + U_3 U_4 U_5 U_6 U_7 U_8 U_9 U_{10} U_{11} - U_1 *$$

$$U_2 U_3 U_4 U_5 U_7 U_8 U_9 U_{10} U_{11} + U_1 U_2 U_3 U_4 U_5 U_6 U_7 U_8 U_9 U_{10} U_{11} + U_2 U_3 U_4 U_5 U_7 U_8 U_9 U_{10} U_{11} - U_2 U_3 U_4 U_5 U_6 U_7 U_8 U_9 U_{10} U_{11} + U_3 U_4 U_5 U_7 U_8 U_{10} U_{11} *$$

$$+ U_1 U_3 U_4 U_5 U_6 U_7 U_8 U_{10} U_{11} + U_2 U_3 U_4 U_5 U_6 U_7 U_8 U_{10} U_{11} - U_1 U_2 U_3 U_4 U_5 U_6 U_7 U_8 U_{10} U_{11} - U_3 U_4 U_5 U_6 U_7 U_8 U_{10} U_{11} - U_1 U_3 U_4 U_5 U_7 U_8 U_{10} U_{11} *$$

$$- U_2 U_3 U_4 U_5 U_7 U_8 U_{10} U_{11} + U_1 U_2 U_3 U_4 U_5 U_7 U_8 U_{10} U_{11} - U_4 U_5 U_7 U_9 U_{10} U_{11} + U_4 U_5 U_6 U_7 U_9 U_{10} U_{11} + U_4 U_5 U_7 U_8 U_9 U_{10} U_{11} - U_4 U_5 U_6 U_7 U_8 U_9 *$$

$$U_{10} U_{11} - U_4 U_5 U_7 U_8 U_{10} U_{11} + U_4 U_5 U_6 U_7 U_8 U_{10} U_{11} - U_2 U_4 U_5 U_7 U_{10} U_{11} + U_1 U_2 U_4 U_5 U_7 U_{10} U_{11} - U_3 U_4 U_5 U_7 U_{10} U_{11} + U_1 U_3 U_4 U_5 U_7 U_{10} U_{11} *$$

$$+ U_2 U_3 U_4 U_5 U_7 U_{10} U_{11} - U_1 U_2 U_3 U_4 U_5 U_7 U_{10} U_{11}$$

HISTORICAL INFORMATION FOR EACH COMPONENT IN MODULE B

COMPONENT	TYPE	SUCCESS (BETA) EQUIVALENT MISSIONS(GAMMA)	FAILURES	PRIOR DISTRIBUTION MEAN
1	TIME-TO-FAILURE(GAMMA)	96.50	2.00	0.969851
2	ATTRIBUTE(BETA)	99.00	1.00	0.980392
3	TIME-TO-FAILURE(GAMMA)	96.50	2.00	0.969851
4	ATTRIBUTE(BETA)	99.00	1.00	0.980392
5	TIME-TO-FAILURE(GAMMA)	96.50	2.00	0.969851
6	ATTRIBUTE(BETA)	99.00	1.00	0.980392
7	TIME-TO-FAILURE(GAMMA)	96.50	2.00	0.969851
8	ATTRIBUTE(BETA)	99.00	1.00	0.980392
9	TIME-TO-FAILURE(GAMMA)	96.50	2.00	0.969851
10	ATTRIBUTE(BETA)	99.00	1.00	0.980392
11	TIME-TO-FAILURE(GAMMA)	96.50	2.00	0.969851
12	ATTRIBUTE(BETA)	99.00	1.00	0.980392
13	TIME-TO-FAILURE(GAMMA)	96.50	2.00	0.969851

SPARCS :: SYSTEM SIMULATICN RCUTINE

SYSTEM RELIABILITY CALCULATED FROM MEAN COMPONENT RELIABILITIES IS 0.903486; SYSTEM UNRELIABILITY IS 0.096514

AVERAGE SYSTEM RELIABILITY FROM 20 MONTE CARLO TRIALS IS 0.909538 ; AVERAGE SYSTEM UNRELIABILTY IS 0.090462

VARIANCE	0.000625
STANDARD DEVIATICN	0.024996

THE MISSION TIME IS 100.000 HOUR

THE ESTIMATED SYSTEM MTBF BASED UPON MEAN COMPONENT RELIABILITIES IS 9.85000000E+02

THE ESTIMATED SYSTEM MTBF BASED UPON MEAN SYSTEM RELIABILITY 20 MONTE CARLO TRIALS IS 1.05464673E+03

PERCENTILE	RELIABILITY PERCENTILE POINTS	MTBF PERCENTILE POINTS	
5.0 PERCENT	0.863285	6.80221680E+02	HOUR
10.0 PERCENT	C.874934	7.48464844E+02	HOUR
20.0 PERCENT	0.889750	8.56057129E+02	HOUR
25.0 PERCENT	0.893619	8.89078657E+02	HOUR
50.0 PERCENT	0.908336	1.04013916E+03	HOUR
75.0 PERCENT	0.927487	1.32843164E+03	HOUR
80.0 PERCENT	C.929598	1.36979785E+03	HOUR
90.0 PERCENT	0.943088	1.70660840E+03	HOUR
95.0 PERCENT	C.948863	1.90508569E+03	HOUR
97.5 PERCENT	0.951640	2.01741943E+03	HOUR
99.0 PERCENT	0.953307	2.09123511E+03	HOUR

It is both interesting and instructive to compare the results of the two outputs, as shown particularly in the assessment tables. The medians or 50-th percentiles are nearly identical, .904296 in Case 6 and .908336 in Case 1. For both problems, percentiles below the 80th are very close to each other. For specified percentiles above the 80th, Case 1 gave higher assessments. Since each output is based on only 20 Monte Carlo trials, these differences do not seem to be excessive. Even closer results would be and have been obtained with, say, 100 Monte Carlo trials.

OPERATING INSTRUCTIONS

The operating instructions for the SPARCS-2 program included in this manual are specific for the IBM series 370/158 (MVS environment) computer system in operation at the University Computer Center at Oklahoma State University. The system includes a PL/1 optimizing compiler. The control cards needed at any computer installation will depend upon local practices.

Instructions for Using Source Desk

The following desk structure may be used with the SPARCS-2 source desk. Note that the facsimilies for two of the 80-column control cards are shown split in half.

COMM. | STATEMENT NUMBER | CONT. | FORTRAN S

```
//SPARCS   JØB (nnnnn,sss-ss-ssss),'user
-name',CLASS=c,TIME=t (continuation of JOB card)
/*PASSWØRD PPPP
// EXEC PLIXCLG,PARM.PLI='AG,A,X',REGIØN
.PLI=400K,REGIØN.GØ=400K (continuation of EXEC card)
//PLI.SYSIN DD *

      Source Deck

//GØ.SYSIN DD *

      Data

//
```

Where: "nnnnn" is a five-digit account number.

"sss-ss-ssss" is your social security number.

"AG, A, X" are the options usually run with the source desk to aid in diagnostics and debugging. Specifying these options generates additional printout information about the source desk. Option "AG" lists an aggregate length table; option "A" lists a full attribute and cross-reference table; and option "X" lists a cross-reference table. The options available may vary from one computer center to another.

Instructions for Using Load Module

A compiled copy of the SPARCS-2 program may be stored in a disk library. The compiled version may then be used again without recompilation. The following desk structure stores the program on the disk.

FORTRAN STATEMENT

```
//SPARCS   JØB (nnnnn,sss-ss-ssss),'user
-name',CLASS=c,TIME=t (continuation of job card)
/*PASSWØRD PPPP
// EXEC PLIXCL
//PLI.SYSIN DD *
  .
  .  Copy of source deck to be compiled and
  .  placed in disk library under a specified name
//LKED.SYSLMØD DD DSNAME=library-name,DISP=ØLD
//LKED.SYSIN DD *
 NAME xxxxxxxx(R)
//
```

Where: "xxxxxxxx" is the name specified to store the program.

The following example demonstrates a deck structure necessary to compile a PL/1 program and subroutines and store them in the library designated OSU. ACTXXXXX. SPARCS under the name SPARCS-2.

```
//SPARCS    JØB CARD (same as previous examples)
/*PASSWØRD PPPP
// EXEC PLIXCL
//PLI.SYSIN DD *
     .
     . Source Deck for SPARCS-2
     .
//LKED.SYSLMØD DD DSNAME=ØSU.ACTXXXXX.SP
ARCS, DISP=ØLD (continuation of LKED card)
//LKED.SYSIN DD *
  NAME SPARCS2(R)
//
```

The following deck structure executes the stored SPARCS-2 program from the disk library.

```
//SPARCS    JØB CARD (same as previous examples)
/*PASSWØRD PPPP
// EXEC PGM=SPARCS2,REGIØN=600K
//STEPLIB DD DSNAME=ØSU.ACTXXXXX.SPARCS,DISP=SHR
//SYSIN DD *
     .
     . Data Cards
     .
//SYSPRINT DD SYSØUT=A
//
```

GLOSSARY

General Terminology

assessment: a schedule of system reliability or MTBF values as a function of the confidence coefficients, the component historical success-or-failure data and the system logical configuration, under an appropriate general statistical, probabilistic, and logical model.

Bayesian techniques: reliability assessment of components from prior distributions whose parameters are functions of available data.

Bernoulli process: governs the occurrence of failures for each pass-fail type component; the results of successive trials are independent and the probability of success on every trial is the component reliability.

beta component: success-or-failure attribute component.

beta distribution: for each pass-fail type component, the conjugate prior distribution on R is "beta", with its parameters being functions of prior failures and prior tests.

coherent system (coherence): the "zero" state (0, ..., 0), all components failed, is a system failure state or cut; the "one" state (1, ..., 1), all components succeeding, is a system success state or path; no paths are contained in any cuts, in the sense of partially ordered sets.

conjugate prior distribution: a prior distribution, such as beta or negative-log gamma, which has a similar mathematical form to the distribution describing events, and for which the parameters are sufficient statistics for the prior data.

gamma component: time-to-failure component.

MTBF: the average operating time between failures; defined only for systems with a mission time.

minimal cut: a set of components which, if they are all failed, cause the system to fail, but if any of them function (and all other components function), the system operates; the system is assumed to be coherent.

minimal path: a set of components which, if they are all functioning, permits the system to function, but if any of them fails (and no other components operate), the system fails; the system is assumed to be coherent.

modularization: by breaking the system up into a logical configuration of independently failing subsystems, as used herein, each system can be represented as a configuration of modules, with each module being a logical configuration of independently failing components.

Monte Carlo: beta or gamma random deviates are generated to obtain the component reliability (or unreliability) values which are substituted into the system probability function.

negative-log gamma distribution: for each time-to-failure component, the prior distribution on the component reliability is negative-log gamma, with its parameters being sufficient statistics for the prior data, failures, and testing time (normalized to mission-equivalent units).

Poincaré's theorem: builds the system reliability function recurrently one minimal path at a time from the minimal paths; dually, a system unreliability function can be built up from the minimal cuts; also known as the method of inclusion-exclusion.

Poisson process: assumed for each time-to-failure component; failures of individual components occur at a constant rate independent of prior history.

reliability: the probability that all assigned functions are performed within a predefined time frame and under the specified environment or environments: the probability of system success.

sufficient statistics: summary statistics which provide as much information about a random sample as if the value of every single observation were available.

super modularity: modules with minimal-cut unreliability calculations can be intermixed with those having minimal-path reliability calculations to obtain the system reliability, with no restrictions other than size and capacity limitations.

system: a configuration of modules unlike one another where each module is a configuration of possibly unlike components, with different failure-history data; beta and gamma components can be intermixed with no restriction on their placement.

REFERENCES

[1] Burris, Jimmy L., "Model for the Analysis of the Probabilities of Systems," MBA research report, Department of Administrative Sciences, Oklahoma State University, 1972.

[2] North American Rockwell Corporation Space Division, "Exact Minimal-Path Techniques for Determining System Reliability," Program MFS-16499; available through NASA's COSMIC, University of Georgia, Athens, GA 30601.

[3] North American Rockwell Corporation Space Division, "System for Computing Operational Probability Equations (SCOPE): Version II," Program FMS-24035; available through NASA's COSMIC, University of Georgia, Athens, GA 30601.

[4] Cooley, John W., "Simulation Program for Assessing the Reliability of Complex Systems (SPARCS)," Ph.D. dissertation, Oklahoma State University, April, 1976.

[5] Locks, Mitchell O., "Monte Carlo Bayesian System Reliability- and MTBF-Confidence Assessment," Air Force Flight Dynamics Laboratory, Wright-Patterson AFB, Ohio, AFFDL-TR-75-144.

[6] Brown, J. R. and M. Lipow, "Testing for Software Reliability," TRW Systems Engineering and Integration Division, One Space Park, Redondo Beach, CA 90278, TRW-SS-75-02, January, 1975.

APPENDIX. "SPARCS-2" SOURCE DECK LISTING

PL/I OPTIMIZING COMPILER SPARCS2: PROC OPTIONS (MAIN); PAGE 2

SOURCE LISTING

STMT

```
 1  SPARCS2: PROC OPTICNS (MAIN);
    /*                                                                   */
    /*********************************************************************/
    /*                                                                   */
    /*   SPARCSII (SIMULATION PROGRAM FOR ASSESSING THE RELIABILITIES OF */
    /* COMPLEX SYSTEM - VERSION II) IS A SYSTEM OF PL/1 PROCEDURES TO    */
    /* ASSESS (PROVIDE INTERVAL ESTIMATES FOR) THE RELIABILITY & MTBF OF */
    /* A COMPLEX SYSTEM OF ANY LOGICAL CONFIGURATION. THE COMPONENTS CAN */
    /* BE EITHER ATTRIBUTE OR TIME-TO-FAILURE(TTF), WITH NO RESTRICTION  */
    /* ON THEIR PLACEMENT. BAYESIAN AND MONTE CARLO TECHNIQUES ARE       */
    /* EMPLOYED.  INPUTS ARE MINIMAL STATES FOR SYSTEM AND ITS MODULES   */
    /* AND COMPONENT FAILURE-HISTORY DATA.  MINIMAL STATES ARE MINIMAL   */
    /* PATHS, FOR 'RELIABILITY' ANALYSIS OR  MINIMAL CUTS FOR            */
    /* 'UNRELIABILITY' ANALYSIS.  OUTPUT IS A SCHEDULE OF SYSTEM         */
    /* RELIABILITY AND MTBF VALUES AND ASSOCIATED CONFIDENCE LEVELS.     */
    /* MTBF ANALYSIS IS OPTIONAL.  FOR EACH COMPONENT, THE RAW DATA ARE  */
    /* EXPERIENCED FAILURE HISTORIES: ACCUMULATED SUCCESSES AND FAILURES */
    /* FOR ATTRIBUTES.  EXPERIENCED TESTING TIME (NORMALIZED TO  MISSION */
    /* EQUIVALENT UNITS) AND FAILURES FOR TTF COMPONENTS.  THE SYSTEM    */
    /* CAN CONSIST OF A COMPLEX CONFIGURATION OF UP TO 128 MODULES, EACH */
    /* MODULE BEING A COMPLEX CONFIGURATION OF UP TO 128 COMPONENTS.     */
    /* MODULES WITH MINIMAL-PATH RELIABILITY ANALYSIS CAN OPTIONALLY BE  */
    /* INTERMIXED WITH MODULES HAVING MINIMAL-CUT UNRELIABILITY          */
    /* CALCULATIONS.  THE SYSTEM CAN OPTIONALLY BE A CONFIGURATION OF    */
    /* EITHER MINIMAL PATHS OR MINIMAL CUTS IN THE MODULES.  OUTPUT IS   */
    /* ALWAYS STANDARDIZED TO SYSTEM RELIABILITY AND ALSO MTBF IF REQUES-*/
    /* TED.  DEFAULT LABELLING IS EMPLOYED FOR COMPONENTS AND MODULES IN */
    /* THE PROBABILITY EQUATION.  COMPONENTS ARE NUMBERED SUCCESSIVELY   */
    /* 1,2, ... ,128.  MODULES ARE SEQUENCED ALPHANUMERICALLY A,B,C, ... */
    /* ,Z, A1,B1, ...,Z1,A2,B2, ... ,Z2, A3,B3, ... ,Z3, A4,B4, ... ,X4. */
    /*                                                                   */
    /* REFERENCES:                                                       */
    /*              M.O.LOCKS, MONTE CARLO BAYESIAN SYSTEM RELIABILITY & */
    /*                         MTBF CONFIDENCE ASSESSMENT, TECHNICAL     */
    /*                         REPORT AFFDL-TR-75-144, AIR FORCE FLIGHT  */
    /*                         DYNAMICS LABORATORY, AIR FORCE SYSTEMS    */
    /*                         COMMAND, WRIGHT PATTERSON AIR FORCE BASE, */
    /*                         1975.                                     */
    /*              M.O.LOCKS, RELIABILITY,MAINTAINABILITY, & AVAILABILITY*/
    /*                         ASSESSMENT., HAYDEN BOOK CO., 1972.       */
    /*              K.K.LEE,   REJECTION METHODS FOR GENERATING RANDOM   */
    /*                         DEVIATES & THEIR APPLICATIONS IN SYSTEM   */
    /*                         RELIABILITY, MASTER'S THESIS, COMPUTING & */
    /*                         INFORMATION SCIENCES, O.S.U., 1977.       */
    /*              J.W.COOLEY, SIMULATION PROGRAM FOR ASSESSING THE     */
    /*                         RELIABILITY OF COMPLEX SYSTEMS (SPARCS),  */
```

PL/I OPTIMIZING COMPILER SPARCS2: PROC OPTIONS (MAIN); PAGE 3

STMT

```
    /*                         PH.D. THESIS, OKLAHOMA STATE UNIVERSITY,  */
    /*                         1976.                                     */
    /*             J.P.CHANDLER, RANF-UNIFORMLY DISTRIBUTED PSEUDO-      */
    /*                         RANDOM NUMBERS., UNPUBLISHED DOCUMENT,    */
    /*                         DEPT OF COMPUTING & INFORMATION SCIENCES  */
    /*                         , OKLAHOMA STATE UNIVERSITY, 1970.        */
    /*             J.L.BURRIS, MODEL FOR THE ANALYSIS OF PROBABILITIES   */
    /*                         OF SYSTEMS (MAPS), MBA RESEARCH REPORT,   */
    /*                         OKLAHOMA STATE UNIVERSITY, 1972.          */
    /*                                                                   */
    /*                                                                   */
    /* PROCEDURE NAMES USED:                                             */
    /*                                                                   */
    /*            ALGAMA        : COMPUTES LOG OF GAMMA FUNCTION.        */
    /*            CABTA         : GENERATES BETA DEVIATES.               */
    /*            COMPUTE       : CALCULATES MODULE OR SYSTEM            */
    /*                            RELIABILITY OR UNRELIABILITY.          */
    /*            DATAGEN       : COMPUTES COMPONENT PRIOR               */
    /*                            DISTRIBUTION MEAN RELIABILITIES.       */
    /*            EQGEN         : GENERATES PROBABILITY EQUATIONS.       */
    /*            EQPUT         : PRINTS OUT PROBABILITY EQUATIONS.      */
    /*            GAUSF         : GENERATES (0,1) NORMAL DEVIATES.       */
    /*            HDLINE        : PRINTS OUT SYSTEM IDENTIFICATION.      */
    /*            HISINF        : PRINTS OUT FAILURE-HISTORY DATA FOR    */
    /*                            COMPONENTS AND PRIOR MEANS.            */
    /*            INPUT1,2,3,4  : READS INPUT DATA.                      */
    /*            MEANREL       : COMPUTES MEAN COMPONENT RELIABILITIES*/
    /*                            FROM FAILURE-HISTORY DATA.             */
    /*            RANF          : GENERATES UNIFORM PSEUDORANDOM         */
    /*                            DEVIATES.                              */
    /*            RGAMA         : GENERATES GAMMA DEVIATES.              */
    /*            SORT          : PERFORMS 'SHELL' SORT.                 */
    /*            STAT          : COMPUTES AND PRINTS R AND MTBF         */
    /*                            AVERAGES, VARIANCES & PERCENTILES.     */
    /*                                                                   */
    /*                                                                   */
    /* INPUT VARIABLES:                                                  */
    /*            ATYPE     -  FLAG INDICATING RELIABILITY OR            */
    /*                         UNRELIABILITY COMPUTATION                 */
    /*            MINPTH    -  BINARY VECTORS FOR MINIMAL PATH           */
    /*            NARG      -  SEED FOR RANDOM NUMBER GENERATION         */
    /*            NCOM      -  NUMBER OF COMPONENTS IN SYSTEM            */
    /*            NMOD      -  NUMBER OF MODULES IN SYSTEM               */
    /*            NPATH     -  NUMBER OF MINIMAL STATES IN SYSTEM        */
    /*            SIMNUM    -  NUMBER OF SIMULATIONS TO BE PERFORMED     */
    /*            SYSID     -  SYSTEM IDENTIFICATION                     */
    /*            TIME      -  MISSION TIME                              */
    /*            UNIT      -  TIME UNIT                                 */
    /*                                                                   */
```

STMT

```
/* INPUT DATA FORMAT:                                                      */00000960
/*                                                                         */00000970
/*                         ** SYSTEM WITHOUT MODULES **                    */00000980
/*                                                                         */00000990
/*                                                                         */00001000
/* DATA    NUMBER   COLUMN                 DESCRIPTIONS                    */00001010
/* SET    OF CARDS POSITION                                                */00001020
/*                                                                         */00001030
/* FIRST      1      1 - 80      ALPHANUMERIC SYSTEM IDENTIFICATION.       */00001040
/*                                                                         */00001050
/* SECOND     1      1 -  4      NUMBER OF MONTE CARLO TRIALS.             */00001060
/*                   6 - 14      SEED VALUE FOR RANDOM NUMBER              */00001070
/*                               GENERATION (MUST BE A NON-ZERO            */00001080
/*                               POSITIVE INTEGER, <= 1.0 E+09).  FOR      */00001090
/*                               EXAMPLE, 4527851 .                        */00001100
/*                  16 - 20      SYSTEM MISSION TIME (OPTIONAL).           */00001110
/*                                                                         */00001120
/* THIRD      1      1 -  3      (NUMERIC ZERO DENOTES NO MODULES).        */00001130
/*                   5 -  7      NUMBER OF COMPONENTS IN SYSTEM(N<129)     */00001140
/*                   9 - 13      NUMBER OF SYSTEM MINIMAL STATES           */00001150
/*                               (M<257).                                  */00001160
/*                       14      R FOR SYSTEM RELIABILITY, U FOR           */00001170
/*                               SYSTEM UNRELIABILITY.                     */00001180
/*                  15 - 19      MISSION TIME UNITS. ALPHABETIC            */00001190
/*                               (OPTIONAL).                               */00001200
/*                                                                         */00001210
/* FOURTH     N   FREE FORMAT    ITEM1: '0'B FOR GAMMA (TIME-TO-           */00001220
/*                                      FAILURE)  OR '1'B FOR BETA         */00001230
/*                                      (SUCCESS-FAILURE).                 */00001240
/*                               ITEM2: NUMBER OF MISSIONS (GAMMA) OR      */00001250
/*                                      SUCCESSES (BETA).                  */00001260
/*                               ITEM3: NUMBER OF FAILURES.                */00001270
/*                                                                         */00001280
/* FIFTH          FREE FORMAT    M MINIMAL STATES AS A STRING OF           */00001290
/*                               N-VECTORS SUCH AS '00110'B.               */00001300
/*                               COMPONENTS IN THE MINIMAL STATE ARE       */00001310
/*                               DENOTED BY 1. COMPONENTS NOT IN           */00001320
/*                               MINIMAL STATE BY 0.                       */00001330
/*                                                                         */00001340
/*                                                                         */00001350
/*                           EXAMPLE                                       */00001360
/*                                                                         */00001370
/*                                                                         */00001380
/*      COLUMN    000000000111111111122222222223333333333                  */00001390
/*      POSITION  123456789012345678901234567890123456789                  */00001400
/*                                                                         */00001410
/*      CARD  6   '10'B   '01'B                                            */00001420
/*            5   '1'B    95.  3.                                          */00001430
/*            4   '0'B    99.  1.                                          */00001440
```


STMT

```
/*            3    0   2      2RHOUR                                       */00001450
/*            2   100 4527851   100                                        */00001460
/*            1   TWO COMPONENT - TWO MINIMAL PATH SYSTEM                  */00001470
/*                                                                         */00001480
/*                                                                         */00001490
/*                                                                         */00001500
/*                         ** SYSTEM WITH MODULES **                       */00001510
/*                                                                         */00001520
/*                                                                         */00001530
/* DATA    NUMBER   COLUMN                 DESCRIPTION                     */00001540
/* SET    OF CARDS POSITION                                                */00001550
/*                                                                         */00001560
/* FIRST      1      1 - 80      ALPHANUMERIC SYSTEM IDENTIFICATION.       */00001570
/*                                                                         */00001580
/* SECOND     1      1 -  4      NUMBER OF MONTE CARLO TRIALS.             */00001590
/*                   6 - 14      SEED VALUE FOR RANDOM NUMBER              */00001600
/*                               GENERATION (MUST BE A NON-ZERO            */00001610
/*                               POSITIVE INTEGER, <= 1.0 E+09).  FOR      */00001620
/*                               EXAMPLE, 2365142 .                        */00001630
/*                  16 - 20      SYSTEM MISSION TIME (OPTIONAL).           */00001640
/*                                                                         */00001650
/*                                                                         */00001660
/* THIRD      1      1 -  3      NUMBER OF MODULES IN SYSTEM (K<129).      */00001670
/*                   5 -  7      NUMBER OF COMPONENTS IN SYSTEM.           */00001680
/*                   9 - 13      NUMBER OF MINIMAL STATES IN SYSTEM        */00001690
/*                               (J<257).                                  */00001700
/*                       14      R FOR SYSTEM RELIABILITY, U FOR           */00001710
/*                               SYSTEM UNRELIABILITY.                     */00001720
/*                  15 - 19      MISSION TIME UNITS. ALPHABETIC            */00001730
/*                               (OPTIONAL).                               */00001740
/*                                                                         */00001750
/* FOURTH         FREE FORMAT    J SYSTEM MINIMAL STATES AS A STRING       */00001760
/*                               OF BINARY K-VECTORS, SUCH AS '010'B.      */00001770
/*                               MODULES IN MINIMAL STATE ARE DENOTED      */00001780
/*                               BY 1, MODULES NOT IN MINIMAL STATE        */00001790
/*                               BY 0.                                     */00001800
/*                                                                         */00001810
/* FIFTH                 1       R FOR MODULE RELIABILITY, U FOR           */00001820
/*                               MODULE UNRELIABILITY.                     */00001830
/*                   2 -  7      MODULE IDENTIFICATION, ALPHANUMERIC.      */00001840
/*                  12 - 14      NUMBER OF COMPONENTS IN MODULE(N<129)     */00001850
/*                  16 - 18      NUMBER OF MINIMAL STATES IN MODULE        */00001860
/*                               (M<257).                                  */00001870
/*                                                                         */00001880
/* SIXTH      M   FREE FORMAT    ITEM1: '0'B FOR GAMMA(TIME-TO-            */00001890
/*                                      FAILURE) OR '1'B FOR BETA          */00001900
/*                                      (SUCCESS - FAILURE).               */00001910
/*                               ITEM2: NUMBER OF MISSIONS(GAMMA) OR       */00001920
/*                                      SUCCESSES(BETA).                   */00001930
```

STMT

```
      /*                                ITEM3: NUMBER OF FAILURES.           */
      /*                                                                     */
      /* SEVENTH         FREE FORMAT    M MODULE MINIMAL STATES AS A STRING  */
      /*                                OF BINARY N-VECTORS SUCH AS          */
      /*                                '01110'B. COMPONENTS IN MINIMAL STATE*/
      /*                                ARE DENOTED BY 1, COMPONENTS NOT IN  */
      /*                                MINIMAL STATE BY 0.                  */
      /*                                                                     */
      /* * THE FIFTH, SIXTH, AND SEVENTH DATA SETS ARE REPEATED  FOR EACH    */
      /*   MODULE.                                                           */
      /*                                                                     */
      /*                                                                     */
      /*                             EXAMPLE                                 */
      /*                                                                     */
      /*                                                                     */
      /*     COLUMN     000000000111111111122222222223333333333              */
      /*     POSITION   123456789012345678901234567890123456789              */
      /*                                                                     */
      /*     CARD  12  '10'B   '01'B                                         */
      /*           11  '1'B    90.  1.                                       */
      /*           10  '0'B    92.  7.                                       */
      /*            9  RB    2  2                                            */
      /*            8  '01'B   '10'B                                         */
      /*            7  '0'B  95.  5.                                         */
      /*            6  '1'B  99.  1.                                         */
      /*            5  UA    2  2                                            */
      /*            4  '0'B    '10'B                                         */
      /*            3    2   2      2RCYCLE                                  */
      /*            2    20 2365142    50                                    */
      /*            1  TWO MODULES AND TWO COMPONENTS SYSTEM ANALYSIS        */
      /*                                                                     */
      /*                                                                     */
      /*                                                                     */
      /***********************************************************************/
      /*                                                                     */
      /*                                                                     */
 2         DCL (A(128),B(128),CREL(128),STMREL,AVSM,AA       ) FLOAT DEC(6);
 3         DCL (TOT,TOT2,VAR,SD,SYSVEC(1000),TIME,MODREL,STMCR) FLOAT DEC(6);
 4         DCL (NCOM,NMOD,NPATH,NTERM,SIMNUM,J,K,NSTRM,M,I,L ) FIXED BIN(15);
 5         DCL COEF(1000) FIXED BIN( 4),SOEF( 500) FIXED BIN(4);
 6         DCL (IS,IT,NCOMS,NSTAT,MSTAT,N                      ) FIXED BIN(15);
 7         DCL (MINPTH(128),TERMS(3500)                        ) BIT (128) VAR;
 8         DCL (SMINP(128),STERM( 500)                         ) BIT(128) VAR;
 9         DCL ATYPE CHAR (1),TYPE (128) BIT(1),UNIT CHAR(5),EQID CHAR (6);
10         DCL SYSID CHAR(79),ATYP CHAR(1);
11         DCL NARG FIXED BIN(31);
12            DCL(A2( 28,128),B2( 28,128),RELMOD(128),RMOL(128))FLOAT DEC(6);
13            DCL(NCOM1(128),MST(128),NTRM( 500)             ) FIXED BIN(15);
14            DCL TYPE2( 28,128) BIT(1),EQID1(128) CHAR(6),ATYP1(128) CHAR(1)
```


STMT

```
                 ,TCOF ( 28,128 ) FIXED BIN(4), TTRM( 28,128 ) BIT(128) VAR;
15    AGN:ON ENDFILE (SYSIN) GO TO FINIR;
16         AVSM = 0.;  TOT = 0.;  TOT2 = 0.;
19         EQID='SYS   '; ATYP=' ';
21    CALL  INPUT1 (ATYPE,NCOM,NMOD,NPATH,SIMNUM,SYSID,TIME,UNIT,NARG);
22         AA=RANF(NARG);
23    IF ¬(NMOD=0) THEN GO TO MODULE;
24    CALL INPUT2 (NCOM,NPATH,TYPE,A,B,MINPTH);
25    CALL MEANREL (NCOM,A,B,TYPE,CREL,ATYPE);
26    CALL HOLINE (SYSID,NMOD,NCOM,NPATH,ATYPE,SIMNUM);
27    CALL MISINF (NCOM,A,B,TYPE,CREL,EQID,ATYPE);
28    CALL EQGEN (COEF,MINPTH,NCOM,NPATH, NTERM,TERMS);/* EQUATION  DATA  */
29    CALL EQPUT (ATYPE,COEF,MINPTH,NCOM,NPATH,NTERM,TERMS,EQID,NMOD,ATYP);
30    CALL COMPUTE (CREL,COEF,NCOM,NTERM,TERMS,STMREL);
31         PUT FILE(SYSPRINT) EDIT ('SPARCS :: SYSTEM SIMULATION ROUTINE')
               (PAGE,SKIP(3),A);
32         IF ATYPE = 'U' THEN STMREL=1.-STMREL;
33         STMCR=STMREL;
34         PUT FILE(SYSPRINT) EDIT (' SYSTEM RELIABILITY CALCULATED FROM ',
            'MEAN COMPONENT RELIABILITIES  IS ',STMREL,': SYSTEM ',
             'UNRELIABILITY IS ',1.-STMREL)(SKIP(5),A,A,F(8,6),A,A,F(8,6));
35   DO IS = 1 TO SIMNUM;                               /* DO SIMULATION     */
36    CALL DATAGEN (A,B,NCOM,TYPE,CREL,ATYPE);
37    CALL COMPUTE (CREL,COEF,NCOM,NTERM,TERMS,STMREL);/*STM RELIABILITY  */
38         IF ATYPE = 'U' THEN STMREL=1.-STMREL;
39         SYSVEC(IS)=STMREL;
40         AVSM = AVSM + STMREL / SIMNUM;
41         TOT = TOT + STMREL;
42         TOT2 = TOT2 + STMREL*STMREL;
43   END;                                            /* END SIMULATION    */
44         GO TO SORTSTAT;
     /*                                                                   */
     /*                         SYSTEM WITH MODULES                       */
     /*                                                                   */
45    MODULE:NCOMS=NCOM;
46         NSTAT=NPATH;
47         CALL INPUT3 (SMINP,NSTAT);
48         CALL HOLINE (SYSID,NMOD,NCOM,NPATH,ATYPE,SIMNUM);
49         CALL EQGEN (SOEF,SMINP,NCOMS,NSTAT,NSTRM,STERM);
50    CALL EQPUT(ATYPE,SOEF,SMINP,NCOMS,NSTAT,NSTRM,STERM,EQID,NMOD,ATYP);
51        DO L=1 BY 1 TO NMOD;
52         CALL INPUT4 (NCOM,MSTAT,TYPE,A,B,MINPTH,ATYP,EQID);
53             DO J=1 TO NCOM;
54              TYPE2(L,J) = TYPE(J);
55              A2(L,J) = A(J);
56              B2(L,J) = B(J);
57             END;      /* NCOM */
58              NCOM1(L)=NCOM;
59              EQID1(L)=EQID;
```

```
STMT

 60             ATYP1(L) = ATYP;
 61             MST(L) = MSTAT;
 62             CALL EOGEN (COEF,MINPTH,NCOM,MSTAT,NTERM,TERMS);
 63             NTRM(L)=NTERM;
 64            DO M=1 TO NTERM;
 65             TCOF(L,M)=COEF(M);
 66             TTRM(L,M)=TERMS(M);
 67            END;     /* NTERM */
 68    CALL EOPUT(ATYPE,COEF,MINPTH,NCOM,MSTAT,NTERM,TERMS,EOID,NMOD,ATYP);
 69    CALL MEANREL (NCOM,A,B,TYPE,CREL,ATYP);
 70        CALL COMPUTE (CREL,COEF,NCOM,NTERM,TERMS,RMOL(L));
 71     IF ATYPE='R' & ATYP='U' THEN RMOL(L)=1.-RMOL(L);
 72     IF ATYPE='U' & ATYP='R' THEN RMOL(L)=1.-RMOL(L);
 73        CALL MISINF (NCOM,A,B,TYPE,CREL,EOID,ATYP);
 74       END;     /* NCOM */
 75        PUT FILE(SYSPRINT) SKIP(5);
 76        CALL COMPUTE (RMOL,SOEF,NCOMS,NSTRM,STERM,STMREL);
 77        PUT FILE(SYSPRINT) EDIT ('SPARCS :: SYSTEM SIMULATION ROUTINE')
               (PAGE,SKIP(3),A);
 78        IF ATYPE = 'U' THEN STMREL=1.-STMREL;
 79        STMCR=STMREL;
 80        PUT FILE(SYSPRINT) EDIT (' SYSTEM RELIABILITY CALCULATED FROM ',
            'MEAN COMPONENT RELIABILITIES IS ',STMREL,': SYSTEM ',
             'UNRELIABILITY IS ',1.-STMREL)(SKIP(5),A,A,F(8,6),A,A,F(8,6));
 81        DO IS=1 TO SIMNUM;
 82          DO N=1 TO NMOD;
 83            NCOM=NCOM1(N);
 84            DO J=1 TO NCOM;
 85              A(J) = A2(N,J);
 86              B(J)=B2(N,J);
 87              TYPE(J)=TYPE2(N,J);
 88            END;  /* NCOM */
 89        CALL DATAGEN (A,B,NCOM,TYPE,CREL,ATYP1(N));
 90          DO M=1 TO NTRM(N);
 91            COEF(M) = TCOF(N,M);
 92            TERMS(M)=TTRM(N,M);
 93          END;  /* NTRM(N) */
 94          CALL COMPUTE (CREL,COEF,NCOM,NTRM(N),TERMS,MODREL);
 95     IF ATYPE='U' & ATYP1 (N)='R' THEN MODREL=1.-MODREL;
 96     IF ATYPE='R' & ATYP1 (N)='U' THEN MODREL=1.-MODREL;
 97          RELMOD(N) = MODREL;
 98          END; /* NMOD */
 99        CALL COMPUTE (RELMOD,SOEF,NCOMS,NSTRM,STERM,STMREL);
100        IF ATYPE = 'U' THEN STMREL=1.-STMREL;
101        SYSVEC(IS)=STMREL;
102        AVSP = AVSM + STMREL / SIMNUM;
103        TOT = TOT + STMREL;
104        TOT2 = TOT2 + STMREL*STMREL;
105        END;                      /*     END SIMULATION WITH MODULES      */
```



```
STMT

106    SORTSTAT:
       CALL SORT (SIMNUM,SYSVEC);
107    CALL STAT(SIMNUM,SYSVEC,AVSM,ATYPE,TIME,UNIT,TOT,TOT2,STMCR);
108        PUT FILE(SYSPRINT) PAGE;
109        GO TO AGN;
      /*                                                                  */
      /*                                                                  */
110    INPUT1: PROC (ATYPE,NCOM,NMOD,NPATH,SIMNUM,SYSID,TIME,UNIT,NARG);
      /********************************************************************/
      /*        THE INPUT1 PROCEDURE READS IN THE DATA FOR THE SYSTEM      */
      /* IDENTIFICATION.                                                  */
      /* ATYPE  - ANALYSIS TYPE (RELIABILITY FOR R, UNRELIABILITY FOR U)   */
      /* NARG   - SEED FOR RANDOM NUMBER GENERATION                       */
      /* NCOM   - NUMBER OF COMPONENTS IN SYSTEM.                         */
      /* NMOD   - NUMBER OF MODULES IN SYSTEM.                            */
      /* NPATH  - NUMBER OF SYSTEM MINIMAL STATES.                        */
      /* SIMNUM - NUMBER OF SIMULATION DESIRED                            */
      /* SYSID  - SYSTEM IDENTIFICATION.                                  */
      /* TIME   - MISSION TIME                                            */
      /* UNIT   - THE UNIT OF TIME MEASUREMENT.                           */
      /********************************************************************/
111        DCL (NCOM,NMOD,NPATH,SIMNUM                    ) FIXED BIN(15);
112        DCL (NARG                                      ) FIXED BIN(31);
113        DCL (TIME                                      ) FLOAT DEC(6);
114        DCL ATYPE CHAR (*),UNIT CHAR(*);
115        DCL SYSID CHAR( *);
116            GET FILE (SYSIN) EDIT (SYSID) (COL(1),A(79)); /* READ CARD 1 */
117            GET FILE (SYSIN) EDIT (SIMNUM,NARG,TIME)
                              (COL(1),F(4),X(1),F(9),X(1),F(6));
118            GET FILE (SYSIN) LIST (NMOD,NCOM,NPATH); /*NEXT 2 READ CARD 3*/
119            GET FILE (SYSIN) EDIT (ATYPE,UNIT) (COL(14),A(1),A(5));
120        RETURN;
121        END  INPUT1;
      /*                                                                  */
      /*                                                                  */
122    INPUT2: PROC (NCOM,NPATH,TYPE,A,B,MINPTH);
      /********************************************************************/
      /* THIS INPUT2 PROCEDURE READS IN THE HISTORICAL DATA FOR COMPONENTS */
      /* AND MINIMAL STATES IN BINARY REPRESENTATION FOR THE COMPUTATION   */
      /* OF SYSTEM RELIABILITY WITHOUT MODULES.                           */
      /* TYPE(*)  - TYPE OF COMPONENT (SEE DATAGEN PROCEDURE).             */
      /* A(*)       - NUMBER OF OBSERVED SUCCESSES (SEE PORT IN DATAGEN).  */
      /* B(*)       - NUMBER OF OBSERVED FAILURES (SEE FAIL IN DATAGEN).   */
      /* MINPTH(*)- MINIMAL STATES IN BINARY REPRESENTATION.              */
      /********************************************************************/
123        DCL (NCOM,NPATH,I                              ) FIXED BIN(15);
124        DCL (A(*),B(*)                                 ) FLOAT DEC (6);
125        DCL MINPTH(*) BIT  (*) VAR , TYPE(*) BIT(*);
126         GET FILE(SYSIN) LIST ((TYPE(I),A(I),B(I) DO I=1 TO NCOM));
```

```
STMT

127        GET FILE (SYSIN) LIST ((MINPTH(I) DO I=1 TO NPATH));
128        RETURN;
129        END  INPUT2;
     /*                                                                  */
     /*                                                                  */
130  INPUT3 : PROC (SMINP,NSTAT);
     /*******************************************************************/
     /* INPUT3 READ IN THE MINIMAL STATES (SMINP) FOR SYSTEM WHICH WILL  */
     /* HAVE THE MODULES IN IT.                                          */
     /*******************************************************************/
131        DCL SMINP(*) BIT(*) VAR, NSTAT FIXED BIN(15);
132        DCL (I                                         ) FIXED BIN(15);
133        GET FILE (SYSIN) LIST ((SMINP(I) DO I=1 TO NSTAT));
134        RETURN;
135        END; /* INPUT3 */
     /*                                                                  */
     /*                                                                  */
136  INPUT4: PROC (NCOM,STATES,TYPE,A,B,MINPTH,ATYP,EQID);
     /*******************************************************************/
     /*      THE  INPUT4  PROCEDURE READS IN THE DATA FOR THE MODULE     */
     /* IDENTIFICATION.                                                  */
     /* ATYP   - ANALYSIS TYPE (RELIABILITY FOR R, UNRELIABILITY FOR U)  */
     /* EQID   - MODULE IDENTIFICATION                                   */
     /* NCOM   - NUMBER OF COMPONENTS IN SYSTEM.                         */
     /* STATES - NUMBER OF SYSTEM MINIMAL STATES.                        */
     /*******************************************************************/
137        DCL (NCOM,STATES) FIXED BIN(15), (A(*),B(*)) FLOAT DEC(6);
138        DCL MINPTH(*) BIT(*) VAR, EQID CHAR(*), ATYP CHAR(*), TYPE(*)
              BIT(*);
139        GET FILE (SYSIN) EDIT (ATYP,EQID) (COL(1),A(1),A(6));
140        GET FILE (SYSIN) LIST (NCOM,STATES);
141        GET FILE (SYSIN) LIST ((TYPE(I),A(I),B(I)  DO I=1 TO NCOM));
142        GET FILE (SYSIN) LIST ((MINPTH(I) DO I=1 TO STATES));
143        RETURN;
144        END: /* INPUT4 */
     /*                                                                  */
     /*                                                                  */
145   MEANREL: PROC (NCOM,A,B,TYPE,ACR,ATY);
     /*******************************************************************/
     /* A COMPUTATION PROCEDURE FOR THE MEAN RELIABILITY OF COMPONENT    */
     /* BASED UPON THE HISTORICAL DATA OF EACH COMPONENT RELIABILITY.    */
     /*******************************************************************/
146        DCL (NCOM,I) FIXED BIN(15),(A(*),B(*),ACR(*)) FLOAT DEC(6),
              TYPE(*) BIT(*),ATY  CHAR(*);
147        DO I=1 TO NCOM;
148        IF TYPE(I)='1'B THEN ACR(I)=(A(I)+1.)/(A(I)+B(I)+2.);
                          ELSE ACR(I)=((A(I)+1)/(A(I)+2.))**(B(I)+1.);
150        IF ATY   = 'U' THEN ACR(I) = 1. - ACR(I);
151        END;
```



```
STMT

152        RETURN;
153        END MEANREL;
     /*                                                                  */
     /*                                                                  */
154   HDLINE:PROC(SYSID,NMOD,NCOM,NPATH,ATYPE,SIMNUM);
     /*******************************************************************/
     /* PRINT OUT THE MAJOR HEADINGS FOR KEY VARIABLES AND SYSTEM ID.    */
     /*******************************************************************/
155        DCL SYSID CHAR (*), ATYPE CHAR (*), STA CHAR(5) VAR;
156        DCL (NCOM,NMOD,NPATH,SIMNUM,NC                 ) FIXED BIN(15);
157        IF ATYPE='R' THEN STA='PATHS';
                     ELSE STA='CUTS ';
159        PUT FILE(SYSPRINT) EDIT ('* S P A R C S *')
              (PAGE,A) ('SIMULATION PROGRAM FOR THE ANALYSIS OF THE ',
              'RELIABILITY OF COMPLEX SYSTEMS') (SKIP(1),A,A) ('SYSTEM ',
              'IDENTIFICATION',SYSID) (SKIP(5),A,A,X(13),A)
              ('NUMBER OF SIMULATIONS ',SIMNUM) (SKIP(1),A,X(14),F(3))
              ('NUMBER OF MODULES ',NMOD) (SKIP(1),A,X(17),F(3))
            ('NUMBER OF COMPONENTS ',NCOM) (SKIP(1),A,X(14),F(3))
               ('NUMBER OF MINIMAL ',STA,NPATH) (SKIP(1),A,A,X(12),F(3));
160        IF ATYPE = 'R' THEN DO;
161        PUT FILE(SYSPRINT) EDIT ('TYPE OF ANALYSIS','RELIABILITY')
                                   (SKIP(1),A,X(21),A);
162        END;
163        ELSE DO;
164        PUT FILE(SYSPRINT) EDIT ('TYPE OF ANALYSIS','UNRELIABILITY')
                                   (SKIP(1),A,X(21),A);
165        END;
166        RETURN;
167        END;  /* HDLINE */
     /*                                                                  */
     /*                                                                  */
168   HISINF: PROC (NCOM,A,B,TYPE,ACR,EQID,ATYPE);
     /*******************************************************************/
     /* PROCEDURE HISINF PRINTS THE HISTORICAL INFORMATIONS FOR EACH     */
     /* COMPONENT EITHER IN A SYSTEM OR IN MODULES.                      */
     /*******************************************************************/
169        DCL ATYPE CHAR(*);
170        DCL (A(*),B(*),ACR(*)                           ) FLOAT DEC(6);
171        DCL TYPE(*) BIT(*),EQID CHAR(*),ID CHAR(13) INIT ('') VAR;
172        DCL (NCOM,NC                                    ) FIXED BIN(15);
173        IF EQID = 'SYS   ' THEN ID='SYSTEM';
174                             ELSE ID='MODULE '||EQID;
175        PUT FILE(SYSPRINT) EDIT ('HISTORICAL INFORMATION FOR EACH ',
              'COMPONENT IN ',ID)(SKIP(5),A,A,A)   ('COMPONENT',' TYPE   ',
               'SUCCESS (BETA)','    FAILURES    ','PRIOR DISTRIBUTION')
              (SKIP(2),A,X(7),A,X(13),A,X(7),A,X(6),A)
               ('EQUIVALENT MISSIONS(GAMMA)','MEAN') (SKIP(1),COL(35),A,
                 X(26),A);
```

```
STMT

 176        IF ATYPE = 'R' THEN DO ;                                            00004880
 177       DO NC=1 TO NCOM;                                                     00004890
 178        IF TYPE(NC)='1'B THEN DO;                                           00004900
 179        PUT FILE(SYSPRINT) EDIT (NC,'ATTRIBUTE(BETA)',A(NC),B(NC),ACR(NC))00004910
                (SKIP(2),X(3),F(3),X(7),A,X(12),F(9,2),X(15),F(5,2),X(15),      00004920
 180            F(8,5));      END;                                              00004930
 181        ELSE DO;                                                            00004940
 182        PUT FILE(SYSPRINT) EDIT (NC,'TIME-TO-FAILURE(GAMMA)',A(NC),B(NC),   00004950
                ACR(NC)) (SKIP(2),X(3),F(3),X(7),A,X(9),F(5,2),X(15),F(5,2),    00004960
 183            X(15),F(8,5));   END;                                           00004970
 184       END;                                                                 00004980
 185        END;                                                                00004990
 186        ELSE DO;                                                            00005000
 187       DO NC=1 TO NCOM;                                                     00005010
 188        IF TYPE(NC)='1'B THEN DO;                                           00005020
 189        PUT FILE(SYSPRINT) EDIT (NC,'ATTRIBUTE(BETA)',A(NC),B(NC),          00005030
                                  1.-ACR(NC))                                   00005040
                (SKIP(2),X(3),F(3),X(7),A,X(12),F(9,2),X(15),F(5,2),X(15),      00005050
 190            F(8,6));      END;                                              00005060
 191        ELSE DO;                                                            00005070
 192        PUT FILE(SYSPRINT) EDIT (NC,'TIME-TO-FAILURE(GAMMA)',A(NC),B(NC),   00005080
             1.-ACR(NC)) (SKIP(2),X(3),F(3),X(7),A,X(9),F(5,2),X(15),F(5,2),    00005090
 193            X(15),F(8,6));   END;                                           00005100
 194       END;                                                                 00005110
 195        END;                                                                00005120
 196        RETURN;                                                             00005130
 197        END; /*HISINF*/                                                     00005140
      /*                                                                      */00005150
      /*                                                                      */00005160
 198  EQGEN:PROC(COEF,MINPTH,NCON,NPATH,NTERM ,TERMS);                          00005170
      /***********************************************************************/00005180
      /*      THE PROBABILITY EQUATIONS ARE ENTIRELY GENERATED IN THIS EQGEN*/00005190
      /* PROCEDURE USING POINCARE-S METHOD (INCLUSION-EXCLUSION). FOR A     */00005200
      /* SYSTEM HAVING N MINIMAL STATES, THE PROBABILITY EQUATION HAS A     */00005210
      /* MAXIMUM OF 2**N-1 TERMS. HOWEVER, THE EQUATION GENERATED BY EQGEN */00005220
      /* HAS ONLY A FRACTION OF THE MAXIMUM NUMBER OF TERMS, BECAUSE OF     */00005230
      /* CANCELLATION OF DUPLICATE TERMS. AFTER EACH MINIMAL STATE IS       */00005240
      /* INTRODUCED AND COMBINED WITH THE PREVIOUSLY GENERATED TERMS        */00005250
      /* TO FORM POSSIBLE NEW TERMS THE ACCUMULATION PROCESS IS PERFORMED. */00005260
      /* TERMS THAT HAVE ZERO COEFFICINNTS ARE REMOVED BEFORE THE NEXT      */00005270
      /* MINIMAL STATE IS INTRODUCED.                                       */00005280
      /*                                                                    */00005290
      /* VARIABLES: COEF - COEFFICIENTS IN PROBABILITY EQUATION,            */00005300
      /*            TERMS- TERMS OF PROBABILITY EQUATION IN BIT STRING.     */00005310
      /***********************************************************************/00005320
 199          DCL COEF (*) FIXED BIN( 4);                                       00005330
 200          DCL MINPTH (*) BIT (*) VAR;                                       00005340
 201          DCL(I1,I2,I3,I4,I5,I6,I7,KK1,KDUP,NSUB,NDUP,INC2)FIXED BIN(15);00005350
 202          DCL (NCON, NPATH,NTERM                         ) FIXED BIN(15);00005360
```



```
STMT

 203          DCL TERMS (*) BIT (*) VAR;                                        00005370
 204          TERMS(1)=MINPTH(1);COEF(1)=1; /* 1ST 3 TERMS OF PROB. EQUA.   */00005380
 206          IF NPATH=1 THEN DO;                                               00005390
 207              NTERM=1;  RETURN;  END;                                       00005400
 210          TERMS(2)=MINPTH(2); COEF(2)=1;                                    00005410
 212          TERMS(3)=MINPTH(1) | MINPTH(2);                                   00005420
 213          COEF(3)=-1;                                                       00005430
 214          NTERM=3;                                                          00005440
 215          IF NPATH=2 THEN GO TO ENDEQ;                                      00005450
 216          NSUB=4;                                                           00005460
 217          DO I1 = 3 TO NPATH;           /* DO LOOP1.  REMAINING TERMS   */00005470
 218          TERMS(NSUB)=MINPTH(I1);                                           00005480
 219          COEF(NSUB)=1;                                                     00005490
 220          NSUB=NSUB+1;                                                      00005500
 221          DO I2 = 1 TO NTERM;           /* DO LOOP2 */                      00005510
 222          TERMS(NSUB)=MINPTH(I1) | TERMS(I2);                               00005520
 223          COEF(NSUB)=-COEF(I2);         /* DETERMINE COEFFICIENT        */00005530
 224          NSUB=NSUB+1;                                                      00005540
 225          END;                          /* END LOOP2 */                     00005550
 226          NDUP=0;                       /* ACCUMALATE DUPLICATE TERMS   */00005560
 227          INC2=NSUB-1;                                                      00005570
 228          DO I3=NTERM+2 TO INC2;                                            00005580
 229          DO I4=3 TO I3-1-NDUP;                                             00005590
 230          IF TERMS(I4)¬=TERMS(I3-NDUP) THEN GO TO ENDI4;                    00005600
 231          COEF(I4)=COEF(I4) + COEF(I3-NDUP);                                00005610
 232          IF I3-NDUP=NSUB-1 THEN GO TO SUB;                                 00005620
 233              DO I5=I3-NDUP TO INC2-1-NDUP;                                 00005630
 234              TERMS(I5)=TERMS(I5+1);                                        00005640
 235              COEF(I5)=COEF(I5+1);                                          00005650
 236              END;                                                          00005660
 237      SUB: NSUB=NSUB-1;                                                     00005670
 238          NDUP=NDUP+1;                                                      00005680
 239          GO TO ENDI3;                                                      00005690
 240    ENDI4: END;                                                             00005700
 241    ENDI3: END;                                                             00005710
 242          KDUP=0;        /* REMOVE TERMS WITH ZERO COEFFICIENTS */          00005720
 243          KK1=NSUB-1;                                                       00005730
 244          DO I6=3 TO KK1;                                                   00005740
 245          IF COEF(I6-KDUP)¬=0 THEN GO TO ENDI6;                             00005750
 246          IF I6-KDUP=NSUB-1 THEN GO TO KSUB;                                00005760
 247              DO I7=I6-KDUP TO KK1-1-KDUP;                                  00005770
 248              TERMS(I7)=TERMS(I7+1);                                        00005780
 249              COEF(I7)=COEF(I7+1);                                          00005790
 250              END;                                                          00005800
 251     KSUB: NSUB=NSUB-1;                                                     00005810
 252          KDUP=KDUP+1;                                                      00005820
 253    ENDI6: END;                                                             00005830
 254          NTERM=NSUB-1;                                                     00005840
 255          END;                          /* END LOOP1 */                     00005850
```

STMT

```
256  ENDEQ: END EOGEN;
     /*                                                                       */
     /*                                                                       */
257  EQPUT: PROC (ATYPE,COEF,MINPTH,NCOM,NPATH,NTERM,TERMS,EQID,NMOD,ATYP);
     /*************************************************************************/
     /* PRINT OUT THE SYSTEM RELIABILITY EQUATION BY USING THE BINARY         */
     /* VECTOR TERMS(*) AND COEFFICIENT OF COEF(*) FROM THE FUNCTION          */
     /* PROCEDURE EUGEN.                                                      */
     /*************************************************************************/
258       DCL (K4,K5,K6,K7,LEN,LEN6,NCOM,NMIN,NTERM,NPATH,NM) FIXED BIN(15);
259       DCL SIGN CHAR(3) VAR, STATE CHAR(5) VAR,NMOD FIXED BIN(15);
260       DCL (CHAR1,CHAR2,MINP                           ) CHAR(120) VAR;
261       DCL MINPTH(*) BIT(*) VAR, TERMS (*) BIT (*) VAR;
262       DCL COEF (*) FIXED BIN( 4), ATYPE CHAR(*),EQID CHAR(*),ATYP
              CHAR(*);
263       DCL KOMPS(128) CHAR(3) VARYING INITIAL('1','2','3',
     '4','5','6','7','8','9','10','11','12','13','14','15','16',
     '17','18','19','20','21','22','23','24','25','26','27',
     '28','29','30','31','32','33','34','35','36','37','38',
     '39','40','41','42','43','44','45','46','47','48','49',
     '50','51','52','53','54','55','56','57','58','59','60',
     '61','62','63','64','65','66','67','68','69','70','71',
     '72','73','74','75','76','77','78','79','80','81','82',
     '83','84','85','86','87','88','89','90','91','92','93',
     '94','95','96','97','98','99','100','101','102','103',
     '104','105','106','107','108','109','110','111','112',
     '113','114','115','116','117','118','119','120','121',
     '122','123','124','125','126','127','128') STATIC;
264  DCL C0 CHAR(1), C1 CHAR(2), C2 CHAR(3), C3 CHAR(4);
265          DCL MOOSY(128) CHAR(3) VARYING INIT('A','B','C',
     'D','E','F','G','H','I','J','K','L','M','N','O','P','Q',
     'R','S','T','U','V','W','X','Y','Z','A1','B1','C1','D1',
     'E1','F1','G1','H1','I1','J1','K1','L1','M1','N1','O1',
     'P1','Q1','R1','S1','T1','U1','V1','W1','X1','Y1','Z1',
     'A2','B2','C2','D2','E2','F2','G2','H2','I2','J2','K2',
     'L2','M2','N2','O2','P2','Q2','R2','S2','T2','U2','V2',
     'W2','X2','Y2','Z2','A3','B3','C3','D3','E3','F3','G3',
     'H3','I3','J3','K3','L3','M3','N3','O3','P3','Q3','R3',
     'S3','T3','U3','V3','W3','X3','Y3','Z3','A4','B4','C4',
     'D4','E4','F4','G4','H4','I4','J4','K4','L4','M4','N4',
     'O4','P4','Q4','R4','S4','T4','U4','V4','W4','X4') STATIC;
266       IF EQID = 'SYS   ' THEN DO;
267       IF ATYPE = 'R' THEN DO;
268                           IF NPATH=1 THEN STATE='PATH ';
269                           ELSE STATE='PATHS';
270                           C0='R'; C1='R '; C2='R  '; C3='R   ';
274       END;
275       ELSE DO;
276                           IF NPATH=1 THEN STATE='CUT  ';
```


STMT

```
277                           ELSE STATE='CUTS ';
278                           C0='U'; C1='U '; C2='U  '; C3='U   ';
282       END;
283       PUT FILE(SYSPRINT) EDIT ('SPARCS :: EQUATION GENERATION ROUTINE')
              (PAGE,SKIP(3),A);
284       END;
285       ELSE DO;
286       IF ATYP  = 'R' THEN DO;
287                           IF NPATH=1 THEN STATE='PATH ';
288                           ELSE STATE='PATHS';
289                           C0='R'; C1='R '; C2='R  '; C3='R   ';
293       END;
294       ELSE DO;
295                           IF NPATH=1 THEN STATE='CUT  ';
296                           ELSE STATE='CUTS ';
297                           C0='U'; C1='U '; C2='U  '; C3='U   ';
301       END;
302       END;
303       IF ¬(EQID='SYS   ') THEN PUT FILE(SYSPRINT) EDIT ('THE ',NPATH,
           ' MINIMAL ',STATE,' FOR ','MODULE ',EQID)(SKIP(3),A,F(3),
304         A,A,A,A,A);     ELSE
          PUT FILE (SYSPRINT) EDIT('THE ',NPATH,' MINIMAL ',STATE,' FOR'
                ,' SYSTEM ')          (SKIP(3),A,F(3),A,A,A,A,SKIP(2));
305       PUT FILE(SYSPRINT) SKIP(2);
306     L5:DO K4=1 TO NPATH;
307       MINP='';
308       MINP='<';
309     L6:DO K5=1 TO NCOM;  /* OUTPUT FOR MINIMAL STATES */
310       IF SUBSTR(MINPTH(K4),K5,1)='1'B THEN
            IF EQID='SYS   ' & NMOD ¬= 0 THEN MINP=MINP|| MOOSY(K5)||',';
311                                    ELSE MINP=MINP|| KOMPS(K5)||',';
312     END;  /* L6 */
313         NMIN=LENGTH(MINP);
314         SUBSTR(MINP,NMIN,1)='>';
315         PUT FILE (SYSPRINT) EDIT (MINP)  (SKIP(1),COL(1),A);
316     END;  /* L5 */
317       PUT FILE (SYSPRINT  EDIT ('NUMBER OF TERMS IN EQUATION:',NTERM)
          (SKIP(2),COL(2),A,F(4,0));
318       PUT FILE (SYSPRINT)  SKIP(2);
319         CHAR1='';  /* DETERMINATION OF COMPONENT SYMBOLS FOR OUTPUT */
320         CHAR2=C0;
321         CHAR1=CHAR1||EQID;
322         CHAR2=CHAR2|| '   = ';
323       L7:DO K6=1 TO NTERM;
324         IF COEF(K6) > 0 THEN  SIGN=' + ';
325                         ELSE  SIGN = ' - ';
326         IF K6 = 1 THEN GO TO AG;
327         CHAR2=CHAR2||SIGN;
328         CHAR1=CHAR1||'   ';
```

```
STMT

329      AG: DO K7 = 1 TO NCOM;
330           IF SUBSTR(TERMS(K6),K7,1)  =  '0'B THEN GO TO L50;
331           IF EQID='SYS    ' & NACO ¬= 0 THEN DO;
332           CHAR1=CHAR1||MODSY(K7)||' ';
333           LEN6=LENGTH(KOMPS(K7));
334           END;
335           ELSE DO;
336           CHAR1=CHAR1||KOMPS(K7)  ||' ';
337           LEN6=LENGTH(KOMPS(K7));
338           END;
339           IF LEN6=1 THEN CHAR2=CHAR2||C1;
340                     ELSE IF LEN6=2 THEN CHAR2=CHAR2||C2;
341                                    ELSE CHAR2=CHAR2||C3;
342           LEN=LENGTH(CHAR2);
343           IF LEN>112 & LEN <120 THEN DO;
344             IF K7 ¬= NCOM THEN CHAR2=CHAR2||'+';
345             PUT FILE (SYSPRINT) EDIT (CHAR2) (SKIP(2),COL(3),A);
346             PUT FILE (SYSPRINT) EDIT (CHAR1) (COL(4),A);
347             LEN=0;
348             CHAR1='';
349             CHAR2='';
350           END;
351           ELSE DO;
352                GO TO L50;
353           END;
354      L50:END; /* AG */
355           END; /* L7 */
356           PUT FILE (SYSPRINT) EDIT (CHAR2) (SKIP(2),COL(3),A);
357           PUT FILE (SYSPRINT) EDIT (CHAR1) (COL(4),A);
358           RETURN;
359           END;
    /*                                                                    */
    /*                                                                    */
360 COMPUTE: PROC(REL,COEF,NCOM,NTERM,TERMS,SYSREL );
    /*********************************************************************/
    /*     PROCEDURE COMPUTE CALCULATES VALUES OF THE RELIABILITY OF     */
    /* EACH SYSTEM BY USING THE PROBABILITY EQUATION GENERATED BY THE    */
    /* PROCEDURE EQGEN.                                                   */
    /*                                                                    */
    /* VARIABLES: SYSREL - SYSTEM RELIABILITY.                           */
    /*********************************************************************/
361           DCL (NCOM,NTERM,KA,KB                          ) FIXED BIN(15);
362           DCL (SYSREL,REL(*)                             ) FLOAT DEC(6);
363           DCL (BROD,DREL,DSYSREL                         ) FLOAT DEC(16);
364           DCL TERMS(*) BIT (*) VAR;
365           DCL COEF (*) FIXED BIN( 4);
366           DSYSREL=0.;
367           DO KA = 1 TO NTERM;
368           BROD=1.0;
```



```
STMT

369           DO KB = 1 TO NCOM;
370           IF SUBSTR(TERMS(KA),KB, 1) ¬= '1'B THEN GO TO FIN;
371           DREL=REL(KB);
372           BROD=BROD*DREL;
373      FIN: END;
374           DSYSREL=DSYSREL+BROD*COEF(KA);
375           END;
376           SYSREL=DSYSREL;
377           END COMPUTE;
     /*                                                                   */
     /*                                                                   */
378  DATAGEN:PROCEDURE    (PORT,FAIL,NCOM,TYPE,REL,ATYPE);
           /*****************************************************************/
           /* THIS FUNCTION PROCEDURE OBTAINS VALUES OF COMPONENT          */
           /* RELIABILITY BY CALLING THE BETA(CABTA) OR GAMMA(RGAMA)       */
           /* RANDOM VARIATES GENERATOR  CORRESPONDING TO EACH             */
           /* CHARACTER OF COMPONENT.                                       */
           /*                                                               */
           /*  VARIABLES: FAIL -    NUMBER OF ACCUMULATED  FAILURES         */
           /*             IV -      INDICATOR FOR MTBF COMPUTATION          */
           /*             NCOM -    NUMBER OF COMPONENTS IN A SYSTEM        */
           /*             REL(*) -  VALUE OF COMPONENT RELIABILITY          */
           /*             TYPE    - 0:FOR TIME-TO-FAILURE COMPONENT         */
           /*                       1:FOR SUCCESS-FAILURE COMPONENT         */
           /*             PORT -    NO. OF SUCCESSES FOR TYPE 1             */
           /*                       NO. OF TESTUNITS FOR TYPE 0             */
           /* REQUIRED FUNCTION PROCEDURES - CABTA, RGAMA                   */
           /*****************************************************************/
379        DCL (REL(*),PORT(*),FAIL(*),AA,B,TEMP              ) FLOAT DEC(6);
380        DCL (NCOM,ID                                        ) FIXED BIN(15);
381        DCL TYPE(*) BIT (*),ATYPE CHAR(*);
382        DO ID=1 TO NCOM;
383        IF TYPE(ID) = '1'B THEN GO TO BTVR;
384                       ELSE IF TYPE(ID) = '0'B THEN GO TO GMVR;
385        PUT FILE (SYSPRINT) LIST ('TYPE,PORT,FAIL DESIGNATED IN ERROR');
386        STOP;
387  BTVR:AA=PORT(ID) + 1.;  /* SUCCESS-FAILURE COMPONENT RELIABILITY     */
388        B = FAIL(ID) + 1.;
389        REL(ID) = CABTA(AA,B);
390        IF ATYPE = 'U' THEN REL(ID) = 1. - REL(ID);
391        GO TO NXTT;
392  GMVR:AA=FAIL(ID) + 1.;  /* TIME-TO-FAILURE COMPONENT RELIABILITY     */
393        B = 1./(PORT(ID)+1.);
394        TEMP   = RGAMA(AA) * B   ;
395        REL(ID) = EXP (-TEMP);
396        IF ATYPE = 'U' THEN REL(ID) = 1. - REL(ID);
397  NXTT:END;
398        END DATAGEN;
     /*
```

STMT

```
     /*
399   CABTA:PROC(A,B) RETURNS (FLOAT DEC(6));                              */
     /************************************************************************/
     /*GENERATES RANDOM DEVIATES FROM THE BETA DISTRIBUTION WITH             */
     /*PARAMETERS EQUAL TO A AND B RESPECTIVELY.                             */
     /* THE PROBABILITY DENSITY IS PROPORTIONAL TO                           */
     /*              X**(A-1) * (1-X)**(B-1) .                               */
     /* THE VALUES OF A AND B MUST EACH BE .GE. 1.0 .                        */
     /* IF THE VALUES OF A, OR B, OR BOTH, ARE .EQ. 1.0 THEN THE METHOD      */
     /* USED IS DIRECT MAPPING OF UNIFORM PSEUDO-RANDOM NUMBERS USING        */
     /* THE INVERSE OF THE CUMULATIVE DISTRIBUTION FUNCTION.                 */
     /* THE METHOD USED IF THE VALUES OF A AND B ARE BOTH .GT. 1.0 IS        */
     /* REJECTION FROM A CAUCHY DENSITY TRUNCATED TO BE NONZERO ONLY IN      */
     /* THE INTERVAL (0,1).  PARH AND PARC ARE SAFETY FACTORS ON THE         */
     /* HEIGHT AND CURVATURE OF THE CAUCHY DENSITY AT THE MODE.              */
     /*                                                                      */
     /* J. P. CHANDLER AND K. R. LEE, OKLAHOMA STATE UNIVERSITY    1977      */
     /*                                                                      */
     /************************************************************************/
400       DCL PARH FLOAT DEC(6) INIT(1.1);
401       DCL PARC FLOAT DEC(6) INIT(1.1);
402       DCL PI FLOAT DEC(5) INIT(3.14159);
403       DCL A1SAV FLOAT DEC(6) INIT(-1.);
404       DCL B1SAV FLOAT DEC(6) INIT(-1.);
405       DCL Z FIXED BIN(31) INIT(0);
406       DCL (A1,B1,ALCNST,ALPA,XMODE,XMODEC,CL,A,B,U,T,GCL ) FLOAT DEC(6);
407       DCL (FBL,DT,BJ,BL
                                                         ) FLOAT DEC(6);
408       IF A = 1. & B = 1. THEN RETURN (RANF(Z));
409       A1=A-1.;
410       B1=B-1.;
     /*                      CHECK FOR ILLEGAL VALUES OF A AND B.
     /*                                                                      */
411     IF A1 < 0. | B1 < 0. THEN GO TO L2;                                  */
412      IF B1=0. THEN RETURN(RANF(Z)**(1./A1));
413               ELSE IF A1=0. THEN RETURN(1.-RANF(Z)**(1./B));
     /*
     /*                      TEST TO SEE IF WE MUST RECOMPUTE THE            */
     /*                            VALUES OF CONSTANTS.                      */
414       IF A1=A1SAV & B1=B1SAV THEN GO TO L1;                              */
415       A1SAV=A1;
416       B1SAV=B1;
417       XMODE=A1/(A1+B1);
418       XMODEC=1.-XMODE;
419       ALCNST=ALGAMA(A+B)-ALGAMA(A)-ALGAMA(B);
420       ALPA=PARC*SQRT(2.*XMODE*XMODEC/(A1+B1));
421       CL= LOG(PARH*PI*ALPA)+ALCNST+A1* LOG(XMODE)+B1* LOG(XMODEC);
422       BL=ATAN(-XMODE/ALPA);
423       BU=ATAN(XMODEC/ALPA);
     /*                      NOW USE REJECTION FROM THE CAUCHY DIST.         */
```


STMT

```
424    L1:DT=TAN(BL+(BU-BL)*RANF(Z))*ALPA;
425       T=XMODE+DT;
426       IF T <= 0. | T >= 1. THEN GO TO L1;
427       GCL=CL+ LOG(ALPA/(ALPA**2+DT**2)/PI);
428       FBL=ALCNST+A1* LOG(T)+B1* LOG(1.-T);
429       IF FBL > GCL THEN GO TO L2;
430    L11:U=RANF(Z);
431       IF U <= 0. THEN GO TO L11;
432       IF GCL+LOG(U) > FBL THEN GO TO L1;
433          RETURN(T);
434    L2:PUT FILE(SYSPRINT) LIST ('  ERROR IN SUBROUTINE CABTA  ',A,B,
                          XMODE,ALCNST,ALPA,CL,T,GCL,FBL);
435       STOP;
436       END;
     /*                                                                      */
     /*                                                                      */
437   ALGAMA: PROC (XX) RETURNS (FLOAT DEC(6));
     /************************************************************************/
     /*      COMPUTES THE DOUBLE PRECISION NATURAL LOGARITHM OF THE GAMMA    */
     /*      FUNCTION OF A GIVEN DOUBLE PRECISION ARGUMENT.                  */
     /************************************************************************/
438       DCL   (ZZ,TERM,RZ2)      FLOAT DEC (14),
                (XX,DLNG)          FLOAT DEC (6), ERROR CHAR(1) INIT('0');
439     ZZ    =XX ;
440     IF XX <= 1.E10
        THEN IF XX <= 1.E-09                /* XX IS NEAR 0 OR NEGATIVE    */
             THEN DO ;                      /* SET ERROR INDICATOR         */
441               ERROR='2' ; PUT FILE(SYSPRINT) DATA (ERROR) SKIP(2);
443               DLNG =-1.E75 ;
444               GO TO S20 ;
445               END ;
446          ELSE DO ;                      /* XX > 0 AND < OR = TO 1.E+10 */
447               TERM =1.E0 ;
448  S10:         IF ZZ <= 18.E0            /* ZZ < OR = 18                */
                  THEN DO ;                 /* TRANSLATE ARGUMENT          */
449                    TERM =TERM*ZZ ;
450                    ZZ   =ZZ+1.E0 ;
451                    GO TO S10 ;
452                    END ;
453               ELSE DO ;                 /* CALC. EQUATION 1            */
454                    RZ2  =1.E0/ZZ**2 ;
455                    DLNG =(ZZ-0.5E0)*LOG(ZZ)-ZZ+0.9189385332046727E0
                          -LOG(TERM)+(1.E0/ZZ)*(.8333333333333333E-01
                          -(RZ2*(.2777777777777777E-02+(RZ2*
                          (.7936507936507936E-03-(RZ2*
                          (.5952380952380952E-03)))))))) ;
456                    GO TO S20 ;
457                    END ;
458               END ;
```

```
STMT

 459      ELSE IF XX <  1.E70 THEN DO;         /* XX > 1.E+10 AND < 1.E+70      */00008800
 460               DLNG =ZZ*(LOG(ZZ)-1.E0) ;/* CALC. EQUATION 2                  */00008810
 461               GO TO S20 ;                                                      00008820
 462               END ;                                                            00008830
 463           ELSE DO :                         /* XX > OR = 1.E+70              */00008840
 464               ERROR='1' ;    PUT FILE(SYSPRINT) DATA (ERROR) SKIP(2);          00008850
 466               DLNG =1.E75 ;                                                    00008860
 467               END ;                                                            00008870
 468    S20:RETURN (DLNG); END;                                                     00008880
      /*                                                                           */00008890
      /*                                                                           */00008900
 470  RGAMA: PROC (A) RETURNS (FLOAT DEC(6));                                       00008910
      /***************************************************************************/00008920
      /*GENERATES A RANDOM GAMMA VARIATE WITH PARAMETER -A-, A.GT.1/3.           */00008930
      /*GEORGE MARSAGLIA, SCHOOL OF COMPUTER SCIENCE, MCGILL UNIVERSITY          */00008940
      /*                                                                         */00008950
      /* CODED IN PL/I BY K. K. LEE.                                             */00008960
      /*                                                                         */00008970
      /***************************************************************************/00008980
 471       DCL B FLOAT DEC(6) INIT (-1.);                                           00008990
 472       DCL O FIXED BIN(31) INIT(0);                                             00009000
 473       DCL (A,CL,S,ROOTH ,CS,X,Z,U,E,CD,T,CC,GAMA,ZO .   ) FLOAT DEC(6);         00009010
 474       CL=3.*A-1.;                                                              00009020
 475       IF CL<= 0. THEN STOP; /* TEST FOR INVALID VALUE OF -A-.                */00009030
      /*                            HAS THE VALUE OF -A- CHANGED SINCE THE        */00009040
      /*                            PREVIOUS CALL TO RGAMA.                       */00009050
 476       IF A = B THEN GO TO ONE;                                                 00009060
 477       B=A;                          /*    RECOMPUTE VARIOUS CONSTANTS.       */00009070
 478       S=1./(3.*SORT(A));                                                       00009080
 479       ROOTH=SORT(3.);                                                          00009090
 480       ZO=1.-ROOTH*S;                                                           00009100
 481       CC=A*ZO**3-(S-ROOTH)**2/2.;                                              00009110
 482       CS=1.-S**2;                                                              00009120
 483   ONE:X=GAUSF(O);        /* GAUSF RETURNS A NORMAL(0,1) RANDOM DEVIATE       */00009130
 484       Z=S*X+CS;                                                                00009140
 485       IF Z <= 0. THEN GO TO ONE;                                               00009150
 486       GAMA=A*Z**3 ;                                                            00009160
 487   SIX:U=RANF(O);                                                               00009170
 488       IF U <= 0. THEN GO TO SIX;                                               00009180
 489       E = - LOG(U);          /* GENERATE E, AN EXPONENTIAL RANDOM DEVIATE*/00009190
 490       CD=E+X**2/2.- GAMA+CC;                                                   00009200
 491       T=1.-ZO/Z;                                                               00009210
 492       IF CD+CL*T*(1.+T*(1./2.+T/3.)) > 0. THEN GO TO EGT;                      00009220
 493       IF CD+CL* LOG(Z/ZO) < 0. THEN GO TO ONE;                                 00009230
 494   EGT:RETURN( GAMA);                                                           00009240
 495       END;                                                                     00009250
      /*                                                                         */00009260
      /*                                                                         */00009270
 496   GAUSF:PROC (NDUMY) RETURNS (FLOAT DEC(6));                                   00009280
```



```
STMT

      /***************************************************************************/00009290
      /*GENERATES RANDOM NORMAL DEVIATES...  RANDOM NUMBERS FROM A GAUSSIAN*/00009300
      /*DISTRIBUTION WITH ZERO MEAN AND UNIT VARIANCE.                           */00009310
      /*G. MARSAGLIA AND T. A. BRAY, S.I.A.M. REVIEW 6 (1964) 260.               */00009320
      /*FOR A RANDOM NUMBER FROM THE GAUSSIAN DISTRIBUTION WITH MEAN VALUE */00009330
      /*EQUAL TO 'AMEAN' AND STANDARD DEVIATION EQUAL TO 'SIGMA', USE....  */00009340
      /*        RNG=AMEAN+SIGMA*GAUSF(DUMMY)                                     */00009350
      /*                                                                         */00009360
      /* J. P. CHANDLER, COMPUTING & INFORMATION SCIENCE, O S U                  */00009370
      /* CODED IN PL/I BY K. K. LEE.                                             */00009380
      /*                                                                         */00009390
      /***************************************************************************/00009400
 497       DCL NARG FIXED BIN(31) INIT(0);                                          00009410
 498       DCL NARGA FIXED BIN(31) INIT(0);                                         00009420
 499       DCL NDUMY FIXED BIN(31);                                                 00009430
 500       DCL (A,ABX, EX,FAC,G,GAUS ,SUMSQ,VA,VB,X,Y,TMP  ) FLOAT DEC(6);          00009440
 501       DCL (RJUMP,B                                    ) FLOAT DEC(6);          00009450
 502       GAUS = 0.;                                                               00009460
 503       NARGA=NARG;                                                              00009470
 504       RJUMP=RANF(NARG);                                                        00009480
 505       IF RJUMP < .1362 THEN GO TO L20;                                         00009490
 506       A=(RJUMP-.1362)/.8638;                                                   00009500
      /*                            DEFEAT THE IDIOTIC EMR F4 OPTIMIZATION.   */00009510
 507       B=RANF(NARG);                                                            00009520
 508       TMP=A+B+RANF(NARGA)-1.5;                                                 00009530
 509       GAUS=TMP*2.;                                                             00009540
 510       GO TO L150;                                                              00009550
 511   L20: IF RJUMP < .0255 THEN GO TO L40;                                        00009560
 512       A=(RJUMP-.0255)/.1107;                                                   00009570
 513       GAUS=1.5*(A+RANF(NARG)-1.);                                              00009580
 514       GO TO L150;                                                              00009590
 515   L40: IF RJUMP < .0026997961 THEN GO TO L110;                                 00009600
 516       X=6.*RANF(NARG)-3.;                                                      00009610
 517       Y=.358*RANF(NARGA);                                                      00009620
 518       EX=17.49731196*EXP(-.5*X**2) ;                                           00009630
 519       ABX=ABS(X);                                                              00009640
 520       IF ABX > 1. THEN GO TO L70;                                              00009650
 521       G=EX-4.73570326*(3.-X**2)-2.15787544*(1.5-ABX);                          00009660
 522       GO TO L90;                                                               00009670
 523   L70:G=EX-2.36785163*(3.-ABX)**2;                                             00009680
 524       IF ABX > 1.5 THEN GO TO L90;                                             00009690
 525       G=G-2.15787544*(1.5-ABX);                                                00009700
 526   L90: IF Y > G THEN GO TO L150;                                               00009710
 527       GAUS=X;                                                                  00009720
 528       GO TO L150;                                                              00009730
 529  L110:VA=2.*RANF(NARG)-1.;                                                     00009740
 530       VB=2.*RANF(NARGA)-1.;                                                    00009750
 531       SUMSQ=VA**2+VB**2;                                                       00009760
 532       IF SUMSQ > 1. THEN GO TO L110;                                           00009770
```

```
STMT

533        IF SUMSQ <= 0. THEN GO TO L110;                                     00009780
534        FAC=SQRT((9.-2.* LOG(SUMSQ))/SUMSQ);                                00009790
535        GAUS=VA*FAC ;                                                       00009800
536        IF ABS(GAUS)  < .3 THEN GO TO L110;                                 00009810
537   L150:RETURN(GAUS);                                                       00009820
538        END;                                                                00009830
      /*                                                                     */00009840
      /*                                                                     */00009850
      /*                                                                     */00009860
539    (NOFIXEDOVERFLOW) : RANF : PROC (NARG) RETURNS (FLOAT DEC (6)):         00009870
      /**********************************************************************/00009880
      /* THIS FUNCTION GENERATES PSEUDO-RANDOM NUMBERS, UNIFORMLY            */00009890
      /* DISTRIBUTED ON (0,1).  THIS VERSION IS FOR THE IBM 360.             */00009900
      /* METHOD... COMPOSITE OF THREE MULTIPLICATIVE CONGRUENTIAL GENERATOR*/00009910
      /*     G. MARSAGLIA AND T. A. BRAY, COMM. ACM 11 (1968) 757.           */00009920
      /* IF RANF IS CALLED WITH NARG=0, THE NEXT RANDOM NUMBER IS RETURNED.*/00009930
      /* IF RANF IS CALLED WITH NARG¬=0, THE GENERATOR IS RE-INITIALIZED    */00009940
      /* USING IABS(2*NARG+1) AND THE FIRST RANDOM NUMBER FROM THE NEW      */00009950
      /* SEQUENCE IS RETURNED.                                               */00009960
      /*                                                                     */00009970
      /* CODED IN PL/I BY WILLIAM SCOTT DAVIS FROM A FORTRAN VERSION BY     */00009980
      /* J. P. CHANDLER, COMPUTER SCIENCE DEPT., OKLAHOMA STATE UNIVERSITY.*/00009990
      /*                                                                     */00010000
      /**********************************************************************/00010010
540            DCL K FIXED BINARY(31)   INITIAL(7654321) STATIC;               00010020
541            DCL L FIXED BINARY(31)   INITIAL(7654321) STATIC;               00010030
542            DCL M FIXED BINARY(31)   INITIAL(7654321) STATIC;               00010040
543            DCL MK FIXED BINARY(31)   STATIC INITIAL(282629);               00010050
544            DCL ML FIXED BINARY(31)   STATIC INITIAL(34821);                00010060
545            DCL MM FIXED BINARY(31)   STATIC INITIAL(65541);                00010070
546            DCL NARG FIXED BIN(31);                                         00010080
547            DCL NFIRST BIT(1) STATIC INITIAL('1'B);                         00010090
548            DCL (J,KLM,N(128),NDIV,NR                ) FIXED BIN(31) STATIC;00010100
549            DCL (RAN,RDIV                            ) FLOAT DEC(6) STATIC: 00010110
550            IF NARG ¬= 0 THEN DO:         /* RE-INITIALIZE USING NARG.    */00010120
551                KLM = ABS(2 * NARG + 1);                                    00010130
552                K,L,M = KLM;                                                00010140
553            END;                                                            00010150
554            ELSE DO;                                                        00010160
555            IF ¬ NFIRST THEN GO TO SKP;                                     00010170
556            END;                                                            00010180
557            NFIRST = '0'B;                /* INITIALISE THE ROUTINE.      */00010190
558            NDIV = 16777216;                                                00010200
559            RDIV = 32768.0 * 65536.0;                                       00010210
560            DO J = 1 TO 128;              /* FILL THE TABLE.              */00010220
561                K = K * MK;                                                 00010230
562                N(J) = K;                                                   00010240
563              END;                                                          00010250
564   SKP:     L = L * ML;                   /* COMPUTE THE NEXT RANDOM NO.  */00010260
```



```
STMT

565            J = 1 + ABS(L) / NDIV;                                          00010270
566            M = M * MM;                                                     00010280
567            NR = ABS(N(J) + L + M);                                         00010290
568            RAN = FLOAT(NR) / RDIV;                                         00010300
569            K = K * MK;          /* REFILL THE J-TH PLACE IN THE TABLE     */00010310
570            N(J) = K;                                                       00010320
571            RETURN(RAN);                                                    00010330
572            END RANF;                                                       00010340
      /*                                                                     */00010350
      /*                                                                     */00010360
573   SORT:PROC(N,X);                                                          00010370
      /**********************************************************************/00010380
      /*                                                                     */00010390
      /* SORTING PROCEDURE ON THE VALUES OF THE SYSTEM RELIABILITY BY       */00010400
      /* USING SHELL SORTING METHOD.                                         */00010410
      /*                                                                     */00010420
      /* HIBBARD T. N.  AN EMPIRICAL STUDY OF MINIMAL STORAGE SORTING.      */00010430
      /*                 COMM. ACM 6 (MAY,1963), 206.                        */00010440
      /*                                                                     */00010450
      /**********************************************************************/00010460
574         DCL (N,M,I,J                                       ) FIXED BIN(15);00010470
575         DCL (X(*),Y                                        ) FLOAT DEC(6): 00010480
576         IF N<2 THEN RETURN;                                                00010490
577         M = N;                                                             00010500
578    ONE:M = M/2;                                                            00010510
579         DO J=1 TO N-M;                                                     00010520
580         Y = X(J+M);                                                        00010530
581         DO I=J BY -M TO 1;                                                 00010540
582         IF X(I)<Y THEN GO TO TWO;                                          00010550
583         X(I+M) = X(I);                                                     00010560
584         END;                                                               00010570
585    TWO:X(I+M) = Y;                                                         00010580
586         END;                                                               00010590
587         IF M>1 THEN GO TO ONE;                                             00010600
588         RETURN;                                                            00010610
589         END; /* SHELL SORT */                                              00010620
      /*                                                                     */00010630
      /*                                                                     */00010640
590    STAT: PROC (SN,SV,AVSM,ATYPE,TIME,UNIT,TOT,TOT2,STMCR);                 00010650
      /**********************************************************************/00010660
      /* PROCEDURE STAT COMPUTES THE ESTIMATED SYSTEM MTBF,SYSTEM           */00010670
      /* RELIABILITY AND MTBF PERCENTILE POINTS USING  A  SET OF SYSTEM     */00010680
      /* RELIABILITIES GENERATED BY MONTE CARLO PROCEDURE.                   */00010690
      /**********************************************************************/00010700
591         DCL PCT(11) FLOAT DEC(6) INIT (.05,.1,.2,.25,.5,.75,.8,.9,.95,.97500010710
                ,.99) STATIC,(VAL(11),MTBF(11),SV(*),AVSM,MTBA) FLOAT DEC(6); 00010720
592         DCL (I,SN,IV                                     ) FIXED BIN(15);00010730
593         DCL (TIME,FVAL,CVA,SZR,FLOG,TOT,TOT2,SD,VAR,PR    ) FLOAT DEC(6): 00010740
594         DCL UNIT CHAR(*),ATYPE1 CHAR(13),ATYPE CHAR(*),STMCR FLOAT DEC(6);00010750
```

```
STMT

595      DO I=1 TO 11:   VAL(I)=0.;       MTBF(I)=0.;       END;
599      ATYPE1 = 'RELIABILITY';
600      IF SN=1 THEN DO:
601      VAR = 0.; SD=0.;
603      END;
604      ELSE DO:
605      VAR=(TOT2-TOT*TOT/SN)/(SN-1);
606      SD = SORT(VAR);
607      END;
608      PUT FILE(SYSPRINT) EDIT (' AVERAGE SYSTEM RELIABILITY FROM ',
              SIMNUM,' MONTE CARLO TRIALS IS ',AVSM,' ; AVERAGE SYSTEM ',
              'UNRELIABILTY IS ',1.-AVSM)
              (SKIP(5),A,F(3),A,F(8,6),A,A,F(8,6))
                                    (' VARIANCE                                  *
              ,' ' ,VAR) (SKIP(2),A,A,F(8,6)) (' STANDARD DEVIATION           *
              ,'          ',SD)           (SKIP(2),A,A,F(8,6))   ;
609      IF TIME ¬= 0 THEN
610    DO:MTBMCR=-TIME/LOG(STMCR);
611      MTBA=-TIME/LOG(AVSM):
612      PUT FILE(SYSPRINT) EDIT ('THE MISSION TIME IS ',TIME,UNIT)
              (SKIP(5),COL(6),A,F(8,3),X(2),A) ('THE ESTIMATED SYSTEM MTBF '
              ,'BASED UPON MEAN COMPONENT RELIABILITIES IS ',MTBFMCR)
              (SKIP(2),COL(6),A,A,E(15,8))('THE ESTIMATED SYSTEM MTBF ',
              'BASED UPON MEAN SYSTEM RELIABILITY ',SIMNUM,' MONTE CARLO ',
              'TRIALS IS ',MTBA) (SKIP(2),COL(6),A,A,F(3),A,A,E(15,8))
              (ATYPE1,'MTBF') (SKIP(3),
              COL(24),A,COL(49),A) ('PERCENTILE','PERCENTILE','PERCENTILE')
              (COL(4),A,COL(25),A,COL(46),A) ('POINTS','POINTS') (COL(27),
              A,COL(48),A);
613    END;
614      ELSE DO:
615           PUT FILE (SYSPRINT) EDIT ('PERCENTILE','PERCENTILE') (SKIP(3),
         COL(5),A,COL(29),A) ('POINTS') (COL(31),A);
616      END;
    /*                                                                   */
617      DO I=1 TO 11;
618      VAL(I)=SN*PCT(I)+.5;
619      IV=FIXED(VAL(I));
620      FVAL=FLOAT(IV);
621      IF FVAL ¬= VAL(I) THEN DO:
622      VAL(I)=VAL(I)-FVAL;
623      IF IV=0 THEN DO:
624      SZR=SV(1)*2.-SV(2);
625      CVA=SV(2)-SV(1);
626      VAL(I)=CVA*VAL(I);
627      VAL(I)=SZR+VAL(I);
628      END:
629      ELSE DO;
630      IF IV=SN THEN DO;
```



```
STMT

631      CVA=SV(IV)-SV(IV-1);
632      END;
633      ELSE DO:
634      CVA=SV(IV+1)-SV(IV);
635      END;
636      VAL(I)=CVA*VAL(I);
637      VAL(I)=SV(IV)+VAL(I);
638      END;
639      END;
640      ELSE DO;
641      VAL(I)=SV(IV);
642      END;
643      IF TIME ¬= 0. THEN DO;
644      MTBF(I)=-TIME/LOG(VAL(I));
645      PUT FILE(SYSPRINT) EDIT (PCT(I)*100.,'PERCENT',VAL(I),MTBF(I),
              UNIT)(SKIP(2),COL(1),F(7,1),COL(10),A,COL(26),F( 8,6),X(9),
              E(15,8),X(2),A);
646      END;
647      ELSE DO;
648      PUT FILE (SYSPRINT) EDIT (PCT(I)*100.,' PERCENT',VAL(I))
         (COL(2),F(7,1),A,COL(30),F(8,6));
649      END;
650  EXID:END;
651      END; /* STAT */
    /*                                                                   */
    /*                        END OF ALL PROCEDURES                      */
    /*                                                                   */
652  FINIR:
        END;
```


☆U.S.Government Printing Office: 1978 — 757-080/652